



Assessment of an isogeometric approach with Catmull–Clark subdivision surfaces using the Laplace–Beltrami problems

Zhaowei Liu¹ · Andrew McBride¹ · Prashant Saxena¹ · Paul Steinmann^{1,2}

Received: 21 January 2020 / Accepted: 17 June 2020 / Published online: 15 July 2020
© The Author(s) 2020

Abstract

An isogeometric approach for solving the Laplace–Beltrami equation on a two-dimensional manifold embedded in three-dimensional space using a Galerkin method based on Catmull–Clark subdivision surfaces is presented and assessed. The scalar-valued Laplace–Beltrami equation requires only C^0 continuity and is adopted to elucidate key features and properties of the isogeometric method using Catmull–Clark subdivision surfaces. Catmull–Clark subdivision bases are used to discretise both the geometry and the physical field. A fitting method generates control meshes to approximate any given geometry with Catmull–Clark subdivision surfaces. The performance of the Catmull–Clark subdivision method is compared to the conventional finite element method. Subdivision surfaces without extraordinary vertices show the optimal convergence rate. However, extraordinary vertices introduce error, which decreases the convergence rate. A comparative study shows the effect of the number and valences of the extraordinary vertices on accuracy and convergence. An adaptive quadrature scheme is shown to reduce the error.

Keywords Isogeometric analysis · Finite element method · Catmull–Clark subdivision surfaces · Laplace–Beltrami equation

1 Introduction

Hughes et al. [33] proposed the concept of isogeometric analysis (IGA) in 2005. The early works on IGA [10,18,47] focussed on geometries modelled using Non-Uniform Rational B-Splines (NURBS) as these are widely used in computer aided design (CAD). NURBS can be used to model free-form, two-dimensional curves. However, a NURBS surface is a tensor product surface generated by two NURBS curves, thereby imposing limitations for modelling complex geometries with arbitrary topologies. Complex CAD models are always composed of a number of NURBS patches. These patches are often poorly connected in the design stage. When such models are used for analysis, the unmatched patches must be treated carefully to ensure the geometries are watertight. Furthermore, because NURBS can not be

locally refined, adaptive mesh refinement method cannot be employed. A number of alternative CAD techniques were developed and adopted in IGA to overcome these limitations, including Hierarchical B-splines [29,52], T-splines [11,49], PHT-splines [23,42], THB-splines [13,30] and LR B-splines [24,34]. Some of these recent techniques are being adopted by the engineering design market. However, the majority are the subject of academic research and not widely used in the CAD community. Moreover, computing the basis functions for analysis using these alternative approaches can be expensive. Catmull and Clark [14] developed a bicubic B-spline patch subdivision algorithm for describing smooth three dimensional objects. The use of Catmull–Clark subdivision surfaces to model complex geometries in the animation and gaming industries dates back to 1978. Catmull–Clark subdivision surfaces can be considered as uniform bi-cubic splines which can be efficiently evaluated using polynomials.

In CAD, distortion of regular parametrizations are inevitable and indeed vital when modelling complex geometries. Allowing ‘extraordinary vertices’ ensures that Catmull–Clark subdivision surfaces can be used for modelling complex geometries with arbitrary topology. Cirak et al. [17] implemented Loop subdivision surfaces for solving the Kirchhoff–Love shell formulation. This was the first appli-

✉ Zhaowei Liu
zhaowei.liu@glasgow.ac.uk

¹ Glasgow Computational Engineering Centre, University of Glasgow, Glasgow G12 8LT, UK

² Chair of Applied Mechanics, Friedrich-Alexander Universität Erlangen-Nürnberg, Paul-Gordan-Str. 3, 91052 Erlangen, Germany

cation of subdivision surfaces to engineering problems. Subdivision surfaces have subsequently been used in electromagnetics [19], shape optimisation [6,7], acoustics [15,38] and lattice-skin structures [56].

Catmull–Clark subdivision surfaces face a number of challenges when used for analysis. Many of these have been discussed in the literature, however a unified assessment is lacking. This manuscript provides a clear and concise discussion of the challenges and limitations of Catmull–Clark subdivision surfaces.

Engineering designs often require exact geometries including circles, spheres, tori and cones. However, subdivision surfaces can not capture these geometries exactly. Moreover, there are always offsets between the control meshes and the surfaces. Fitting subdivision surfaces [37] aim to overcome this limitation. Although the fitting subdivision surfaces still can not model arbitrary geometries exactly as they are interpolated using cubic splines, they can approximate the given geometries closely through least-square fitting. Another challenge of subdivision surfaces is that they can model smooth closed manifolds easily but require special treatment to model manifolds with boundaries. A common solution is to introduce ‘ghost’ control vertices to provide bases for interpolating. From the perspective of analysis, the shape functions will span into ‘ghost’ elements [17]. In addition, the spline basis functions do not possess an interpolating property. Thus it is difficult to directly impose Dirichlet boundary conditions. Meshless methods and extended finite element methods have developed strategies to overcome this problem [28,39]. A common strategy is to modify the weak form of the governing equation. Methods include the Lagrangian multiplier method [5], the penalty method [3] and Nitsche’s method [32,43].

Conventional Catmull–Clark subdivision surfaces can not be locally refined. Truncated hierarchical Catmull–Clark subdivision surfaces (THCCS), developed by Wei et al. [54], overcome this limitation. They generalise truncated hierarchical B-splines (THB-splines) to meshes with arbitrary topology. Wei et al. [55] subsequently improved their method using a new basis function insertion scheme and thereby enhanced the efficiency of local refinement. The extraordinary vertices introduce singularities in the parametrisation [41,51]. Catmull–Clark subdivision surfaces have C^2 continuity everywhere except at the surface points related to extraordinary vertices where, as demonstrated by Peters and Reif [45], they possess C^1 continuity. Stam [50] developed a method to evaluate Catmull–Clark subdivision surfaces directly without explicitly subdividing, thus allowing one to evaluate elements containing extraordinary vertices. Although the surface gradients can not be evaluated at the extraordinary vertices, they can be evaluated at nearby quadrature points. Thus, subdivision surfaces can be used as C^1 elements as required, for example, in thin shell theory

[17]. Nevertheless, the evaluation of points around extraordinary vertices of Catmull–Clark surfaces introduces error. The conventional evaluation method repeatedly subdivides the element patch until the target point fall into a regular patch allowing a uniform bi-cubic B-spline patch to be mapped the subdivided element patch. The extraordinary vertex also introduces approximation errors because of the singular parameterisations at extraordinary vertices [40,44]. Stam’s natural parametrisation only can achieve C^0 continuity at extraordinary vertices. Recently Wawrzinek and Polthier [53] introduced a characteristic subdivision finite element scheme that adopted a characteristic reparameterisation for elements with extraordinary vertices. The evaluated limiting surface is at least C^1 everywhere and the numerical accuracy is improved. Zhang et al. [57] optimised the subdivision scheme to improve its approximation properties when used for thin-shell theory.

Using the finite element method to solve the partial differential equations (PDEs) on surfaces dates back to the seminal work by Dziuk [25], which developed a variational formulation to approximate the solution of the Laplace–Beltrami problems on two dimensional surfaces. This method was extended to solve nonlinear and higher-order equations on surfaces by Dziuk and Elliott [26]. Dziuk and Elliott [27] also provided a thorough review on finite element methods for approximating the solution of PDEs on surfaces. Dedner et al. [22] proposed a discontinuous Galerkin (DG) method for solving a elliptic problem with the Laplace–Beltrami operator on surfaces. Adaptive DG [21] and high-order DG [1] methods were also developed for solving PDEs on surfaces. However, the accuracy of these methods depends on the approximation of the mean curvatures of the surfaces. The geometrical error is dominant when conventional Lagrangian discretisation is used to approximate solutions on complex surfaces. Isogeometric discretisation maintains the exact geometry and overcomes this limitation. Dedè and Quarteroni [20] proposed an isogeometric approach for approximating several surface PDEs involving the Laplace–Beltrami operator on NURBS surfaces. Bartezzaghi et al. [9] solved PDEs with high order Laplace–Beltrami operators on surfaces using NURBS based isogeometric Galerkin method. More accurate results are obtained using an IGA approach over the conventional finite element method. Langer et al. [36] present an isogeometric DG method with non-matching NURBS patches allowing the approximation of PDEs on more complex surfaces.

This work presents a thorough and unified discussion of several major issues related to isogeometric Galerkin formulation based on Catmull–Clark subdivision surfaces. The difficulties associated with imposing Dirichlet boundary conditions, the reduction of the approximation power around extraordinary vertices, and the problem of sufficient numerical integration in the element with extraordinary vertices

will be examined and discussed. Previous studies [16,17] on Catmull–Clark subdivision surfaces for analysis introduce ghost degrees of freedom for constructing basis functions in elements at boundaries. We propose a method which modifies the basis functions at boundaries to ensure they are only associated with given control vertices. No additional ghost degrees of freedom are involved. A penalty method is employed to impose Dirichlet boundary conditions. This does not change the size or symmetry of the system matrix and is straightforward to implement. An adaptive quadrature scheme inspired by [35] is presented to increase the integration accuracy for elements with extraordinary vertices. The proposed method can perform isogeometric analysis on complex geometries using Catmull–Clark subdivision discretisations. A test for approximating Poisson’s problem on a square plate is conducted to demonstrate the properties of the method in a simplified setting so as to distill the key features. The approach is also used for solving the Laplace–Beltrami equation which is a benchmark problem for curved manifolds [35,41]. A comparative convergence study is conducted between the Catmull–Clark subdivision method and the conventional finite element method. The effects of the extraordinary vertices and modified bases at boundaries on convergence are examined. Catmull–Clark subdivision surfaces are limiting surfaces generated by successively subdividing given control meshes. They are identical to uniform bi-cubic B-splines. Thus, they have difficulty to represent desired geometries exactly. Here, a least-squares fitting method is used to fit any given geometry using Catmull–Clark subdivision surfaces.

This manuscript first summarises the subdivision algorithm and the evaluation method for Catmull–Clark subdivision surfaces. Then, techniques for using Catmull–Clark for numerical analysis and improving accuracy are presented in Sect. 3. Section 4 presents the Laplace–Beltrami problem and Sect. 5 shows a Galerkin method with Catmull–Clark subdivision surface bases. Section 6 showcases the numerical results.

2 Catmull–Clark subdivision surfaces

There exist a variety of subdivision schemes, but the basic idea is to use a subdivision scheme to generate a smooth surface through a limiting procedure of repeated refinement steps starting from an initial polygonal grid. The Catmull–Clark algorithm can generate curves and surfaces which are identical to cubic B-splines. The algorithms for curves and surfaces are shown in Appendices A.1 and A.2, respectively. This section will briefly introduce the methods for interpolating and evaluating curves and surfaces using the Catmull–Clark subdivision algorithm.

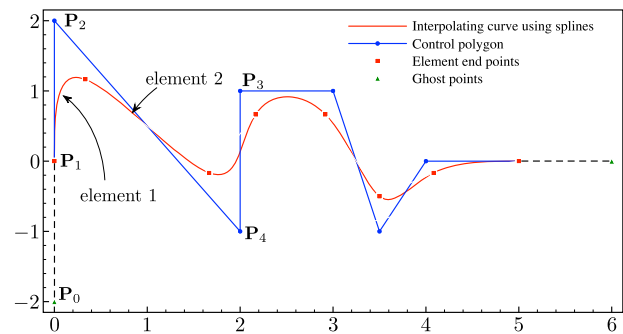


Fig. 1 A subdivision curve is interpolated using basis splines and its control polygon

2.1 Curve interpolation and evaluation based on the subdivision algorithm

Figure 1 shows a curve generated using a subdivision algorithm. The interpolated curve is identical to a cubic B-spline curve. The limiting curve can be interpolated using cubic basis splines and associated control points. With a control polygon containing n control points, the curve is naturally divided into $n - 1$ elements. Each element in the curve is associated with one segment of the control polygon. To interpolate on the target element, four control points including the neighbouring control points are required. For example, if one aims to evaluate the geometry of element 2 in Fig. 1, the four control points P_1, P_2, P_3 and P_4 are required and the curve point is evaluated as

$$\mathbf{x}(\xi) = \sum_{A=1}^4 N_A(\xi) \mathbf{P}_A, \quad (1)$$

where $\xi \in [0, 1]$ is the parametric coordinate within an element. The basis functions for element 2 are defined by

$$\begin{aligned} N_1(\xi) &= \frac{1}{6}[1 - 3\xi + 3\xi^2 - \xi^3], \\ N_2(\xi) &= \frac{1}{6}[4 - 6\xi^2 + 3\xi^3], \\ N_3(\xi) &= \frac{1}{6}[1 + 3\xi + 3\xi^2 - 3\xi^3], \quad N_4(\xi) = \frac{1}{6}\xi^3. \end{aligned} \quad (2)$$

The bases are visualised in Fig. 2a. They are C^2 continuous across element boundaries. Element 1 in Fig. 1 contains the end of the curve, which has an end curve point that coincides with the control point. In order to evaluate this curve, one needs to mirror the point P_2 to P_0 as

$$\mathbf{P}_0 = 2\mathbf{P}_1 - \mathbf{P}_2. \quad (3)$$

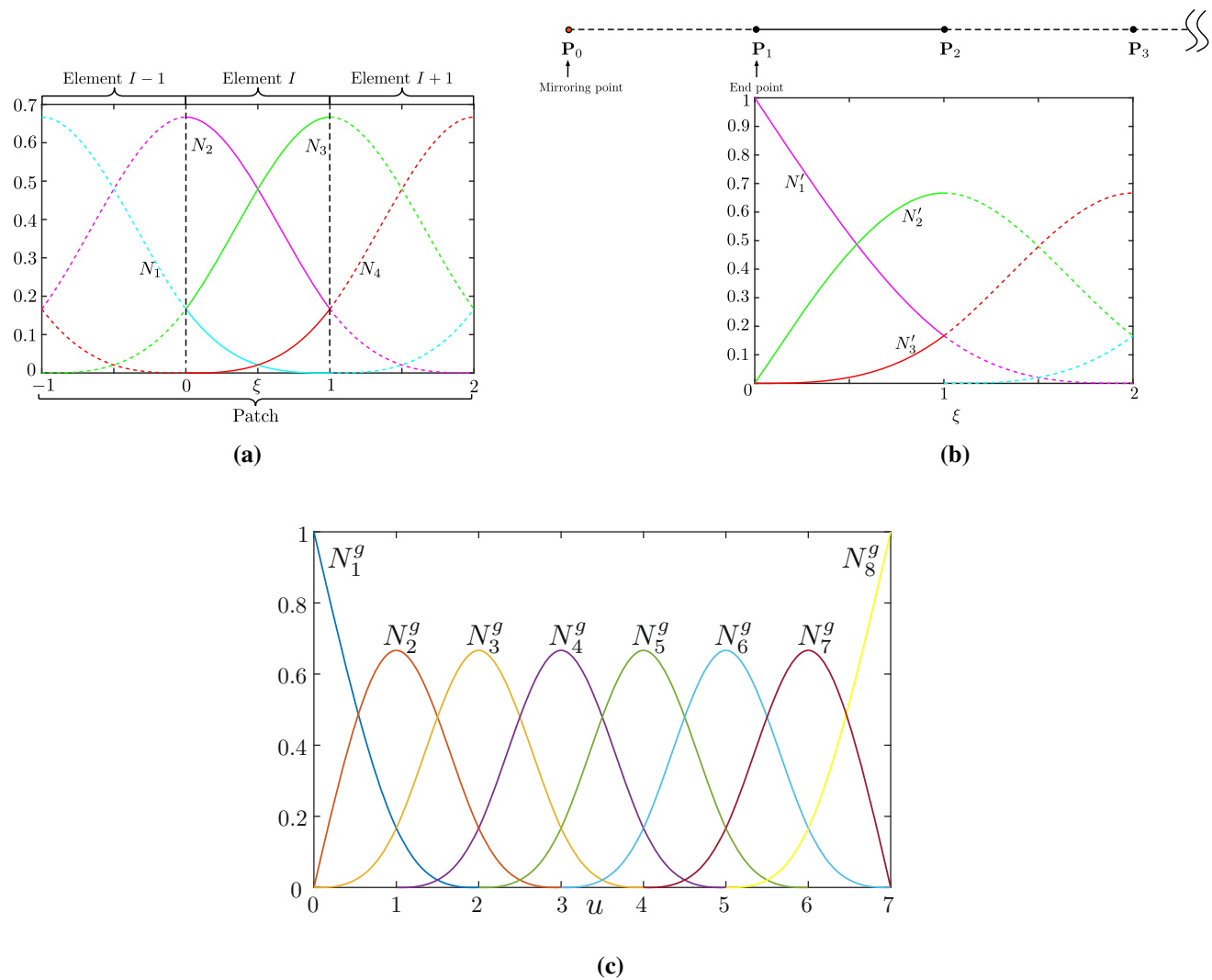


Fig. 2 **a** Basis splines for interpolating element I as a Catmull–Clark curve. **b** The construction of mirroring ghost point to maintain the location of the end point. Basis splines are reconstructed for interpolating an end-element in a Catmull–Clark curve. **c** Global basis functions for interpolating a curve

The curve point can now be evaluated using basis splines with a set of control points shown in Fig. 2b. However, if one adopts a spline discretisation for analysis, this strategy of end element treatment will introduce additional ‘ghost-like’ degrees of freedom. To avoid this problem, the expression for P_0 (3) is substituted into the interpolating equation yielding

$$\mathbf{x}(\xi) = \sum_{A=0}^3 N_{A+1}(\xi) \mathbf{P}_A = \sum_{B=1}^3 N'_B(\xi) \mathbf{P}_B. \quad (4)$$

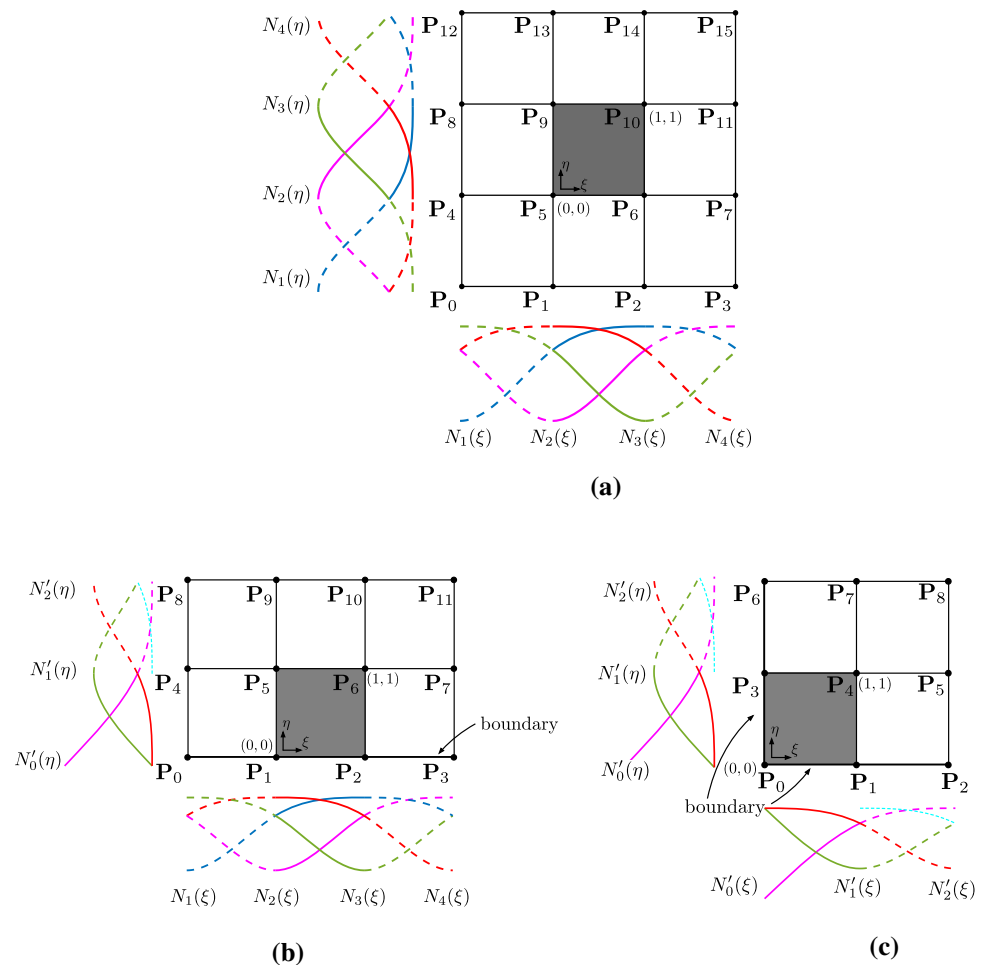
Hence only three control points are required to evaluate a curve point and the modified basis functions for interpolating end elements are defined by

$$\begin{aligned} N'_1(\xi) &= \frac{1}{6}[6 - 6\xi + \xi^3], & N'_2(\xi) &= \frac{1}{6}[6\xi - 2\xi^3], \\ N'_3(\xi) &= \frac{1}{6}\xi^3. \end{aligned} \quad (5)$$

Figure 2b illustrates the modified basis functions. It achieves the same basis functions as the cubic B-Spline with $p + 1$ multiple knots at the two end points. The new basis functions do not possess the Kronecker delta property but do have the interpolating property at the boundary. The performance of modified bases in analysis will be discussed in Sect. 6.1.

The global basis functions for interpolating the curve in Fig. 1 are shown in Fig. 2c. It is worth noting that this subdivision curve is a cubic B-spline curve and represents a special case of Lane–Riesenfeld subdivision it can not model conical shapes exactly. This property is significantly different to NURBS and motivates Sect. 3.1 on geometry fitting.

Fig. 3 Element patches for evaluating a Catmull–Clark subdivision element. **a** A regular element. **b** An element with a face on the boundary. **c** An element with two faces on the boundary



2.2 Interpolating and evaluating Catmull–Clark subdivision surfaces

One defines the number of elements connected with the vertex as the valence. A regular vertex in a Catmull–Clark surface mesh has a valence of 4. A vertex with a valence not equal to 4 is called an extraordinary vertex. This allows subdivision surfaces to handle arbitrary topologies. In their seminal paper [14], Catmull and Clark proposed a way to modify the weight distributions for extraordinary vertices in order to describe complex geometries. With this simple solution, Catmull–Clark surfaces can use a single mesh to present surfaces of arbitrary geometries while other spline-based CAD tools, such as NURBS surfaces, need to link multiple patches. The limiting surface of the Catmull–Clark subdivision algorithm has C^2 continuity over the surface except at the extraordinary vertices where they have C^1 continuity as proven by Peters and Reif [45]. This section will illustrate the methods of interpolating and evaluating both regular element and element with an extraordinary vertex in Catmull–Clark subdivision surfaces.

Element in a regular patch

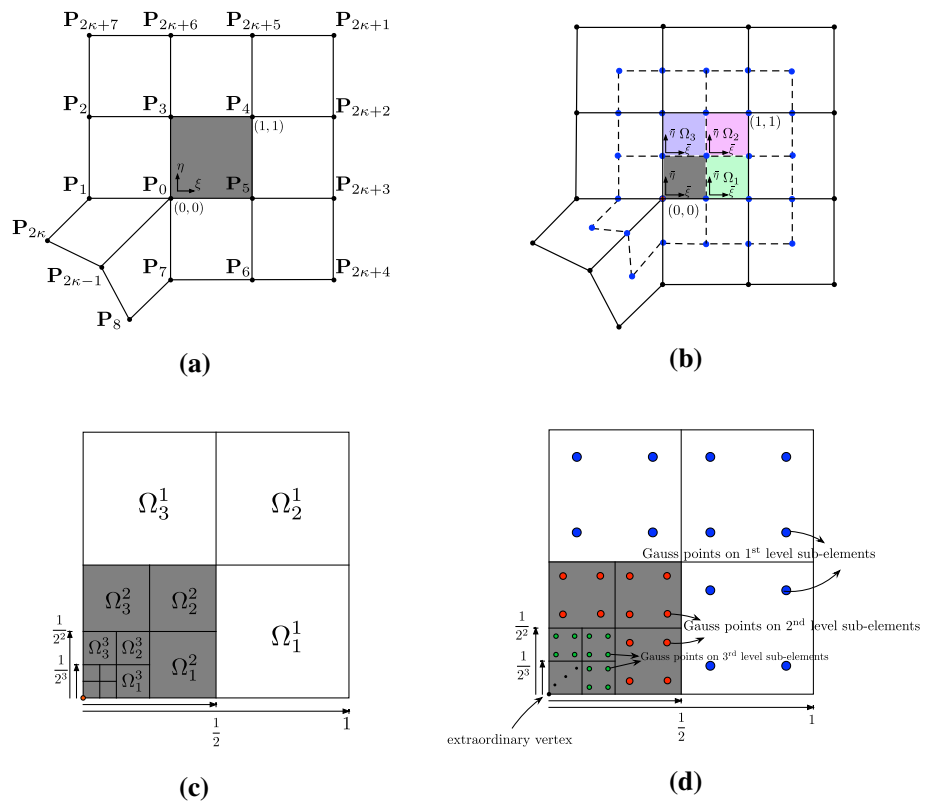
Figure 3a shows a subdivision surface element (dashed) which does not contain an extraordinary vertex. In order to evaluate a point in this Catmull–Clark element, an element patch must be formed. The patch consists of the element itself and the elements which share vertices with it. A regular element patch has 9 elements with 16 control vertices. The surface point can be evaluated using the 16 basis functions associated with these control points as

$$\mathbf{x}(\xi) = \sum_{A=0}^{15} N_A(\xi) \mathbf{P}_A, \quad (6)$$

where $\xi := (\xi, \eta)$ is the parametric coordinate of a Catmull–Clark subdivision surface element. A Catmull–Clark surface is obtained as the tensor product of two Catmull–Clark curves. The basis functions are defined by

$$N_i(\xi) = N_{i \% 4}(\xi) N_{\lfloor i/4 \rfloor}(\eta), \quad i = 0, 1, \dots, 15, \quad (7)$$

Fig. 4 **a** An irregular patch for a Catmull–Clark subdivision element with an extraordinary vertex. **b** One level of subdivision of the element patch divides the element into four sub-elements; three of them are in regular patches. **c** Successive subdivisions of the element until the evaluated point falls into a sub-element with a regular patch. **d** Adaptive Gauss quadrature scheme for the element with an extraordinary vertex



where $N(\xi)$ or $N(\eta)$ are the basis functions defined in Eq. (2) and presented in Fig. 3a. $[\bullet]$ is the modulus operator and $\%$ denotes the remainder operator which gives the remainder of the integer division.

Figure 3b shows the element patch of a subdivision surface element (shaded) that has an edge on the physical boundary. This type of element has only 5 neighbour elements so that it belongs to an element patch which has 12 control vertices. To evaluate this element, a common solution is to generate a set of ‘ghost’ vertices outside the domain to form a full element patch [17]. However, this method involves additional degrees of freedom in numerical analysis. Instead, the curve basis functions in Eq. (5) are adapted to deal with the element on the boundary. The same strategy is used for elements which have two edges on the physical boundary as shown in Fig. 3c.

Element in a patch with an extraordinary vertex

Extraordinary vertices are a key advantage of Catmull–Clark subdivision surfaces which allows them to model complex geometries with arbitrary topologies. However, it increases the difficulty of evaluating the surfaces. Figure 4a shows a Catmull–Clark subdivision element which contains one extraordinary vertex.

In order to evaluate this element, one needs to re-numerate the control points as shown in Fig. 4a. After applying one level of subdivision, new control points are generated and

this element is subdivided into four sub-elements, as shown in Fig. 4b. The sub-elements Ω_1 , Ω_2 and Ω_3 are now in a regular patch. However, the last sub-element (grey) still has an extraordinary vertex. If the target point to be evaluated is in this region, we must repeatedly subdivide the element until the point falls into a sub-element with a regular patch. Then, the point can be evaluated within the sub-element with the new set of control points $\mathcal{P}_{n,k}$, where n is the number of subdivision required and $k = 1, 2, 3$ is the sub-element index shown in Fig. 4b. The new control point set is computed as

$$\mathcal{P}_{n,k} = \mathbf{D}_k \mathbf{A} \bar{\mathbf{A}}^{n-1} \mathcal{P}_0, \quad (8)$$

where \mathbf{D}_k is a selection operator to pick control points for the sub-elements. \mathbf{A} and $\bar{\mathbf{A}}$ are two types of subdivision operators. \mathcal{P}_0 is the initial set of control points. The detailed approach is given in [50] and also can be found in Appendix A.3. $\mathcal{P}_{n,k}$ contains 16 control points. Then, a surface point in the element with an extraordinary vertex can be computed as

$$\mathbf{x}(\xi) = \sum_{A=0}^{15} N_A(\bar{\xi}) \mathbf{P}_A^{n,k}, \quad (9)$$

where $\bar{\xi}$ is the parametric coordinates of the evaluated point in the sub-element, which can be mapped from ξ as

$$\bar{\xi} = \begin{cases} (2^n \xi - 1, 2^n \eta) & \text{if } k = 1 \\ (2^n \xi - 1, 2^n \eta - 1) & \text{if } k = 2 \\ (2^n \xi, 2^n \eta - 1) & \text{if } k = 3 \end{cases}. \quad (10)$$

Equation (9) can thus be rewritten as

$$\mathbf{x}(\xi) = \sum_{A=0}^{2\kappa+7} \hat{N}_A(\xi) \mathbf{P}_A^0, \quad (11)$$

where \hat{N} is the Catmull–Clark subdivision surfaces basis function. Define $\hat{\mathbf{N}}$ as a set of $2\kappa + 8$ basis functions in an element with an extraordinary vertex and \mathbf{N} is a set of 16 regular basis functions defined in Eq. (7). $\hat{\mathbf{N}}$ can be calculated in a vector form as

$$\hat{\mathbf{N}}(\xi) = [\mathbf{D}_k \mathbf{A} \bar{\mathbf{A}}^{n-1}]^T \mathbf{N}(\bar{\xi}). \quad (12)$$

The derivatives of the Catmull–Clark subdivision surfaces basis functions for elements containing extraordinary vertices are expressed as

$$\frac{\partial \hat{\mathbf{N}}(\xi)}{\partial \xi} = \begin{bmatrix} \frac{\partial \hat{N}_0}{\partial \xi} & \frac{\partial \hat{N}_0}{\partial \eta} \\ \frac{\partial \hat{N}_1}{\partial \xi} & \frac{\partial \hat{N}_1}{\partial \eta} \\ \vdots & \vdots \\ \frac{\partial \hat{N}_{2\kappa+7}}{\partial \xi} & \frac{\partial \hat{N}_{2\kappa+7}}{\partial \eta} \end{bmatrix}, \quad (13)$$

and can be computed by

$$\frac{\partial \hat{\mathbf{N}}(\xi)}{\partial \xi} = [\mathbf{D}_k \mathbf{A} \bar{\mathbf{A}}^{n-1}]^T \frac{\partial \mathbf{N}(\bar{\xi})}{\partial \bar{\xi}} \frac{\partial \bar{\xi}}{\partial \xi}, \quad (14)$$

where $\frac{\partial \bar{\xi}}{\partial \xi}$ can be considered as a mapping matrix defined by

$$\frac{\partial \bar{\xi}}{\partial \xi} = \begin{bmatrix} 2^n & 0 \\ 0 & 2^n \end{bmatrix}. \quad (15)$$

Remark 1 The calculation of the basis functions $\hat{\mathbf{N}}$ at a physical point \mathbf{x} involves two mappings. The first is from the physical domain to the parametric domain of an element with an irregular patch, $\mathbf{x} \mapsto \xi$. Because the irregular patch does not have the tensor-product nature, n levels of subdivisions are required and the point is mapped to the parametric domain of a sub-element, $\xi \mapsto \bar{\xi}$. This second mapping is defined in Eq. (10). The value of n approaches positive infinity when ξ approaches the extraordinary vertex which has the parametric coordinate $(0, 0)$. Hence the diagonal terms in the mapping matrix (15) tend to positive infinity as $n \rightarrow \infty$. This results in the basis functions $\hat{\mathbf{N}}$ not being differentiable at $\xi = \mathbf{0}$. This problem is termed singular configuration in [35], and singular parameterisation in [41, 51].

3 Techniques for analysis and improving accuracy

This section presents three techniques which are essential for using Catmull–Clark subdivision surfaces in numerical analysis. A geometry fitting method using Catmull–Clark surfaces is introduced in Sect. 3.1. Section 3.2 illustrates an adaptive quadrature scheme for integrating element with an extraordinary vertex to improve accuracy. Section 3.3 introduces the penalty method for applying essential boundary conditions.

3.1 Geometry fitting

Catmull–Clark subdivision surfaces are CAD tools which can construct limiting surfaces from control polygons and meshes. However, in a number of engineering problems, the geometry is given as an industry design and a limit surface that is a “best approximation” of this desired geometry required. Litke et al. [37] introduced a method for fitting a Catmull–Clark subdivision surface to a given shape. They employed, both a least-squares fitting method and a quasi-interpolation method to determine a set of control points for a given surface. The least-square fitting method is used here. One first chooses a set of sample points $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{n_s}\} \in \Gamma$, where Γ is the geometry, n_s is the number of sample points. Each sample point should be evaluated using Catmull–Clark subdivision bases with control points as

$$\mathbf{s}(\xi) = \sum_{A=1}^{n_b} N_A(\xi) \mathbf{P}_A, \quad (16)$$

where $n_b = 2\kappa + 8$ is the number of local basis functions. Then the set of sample points can be evaluated as

$$\mathcal{S} = \mathbf{L} \mathcal{P}, \quad (17)$$

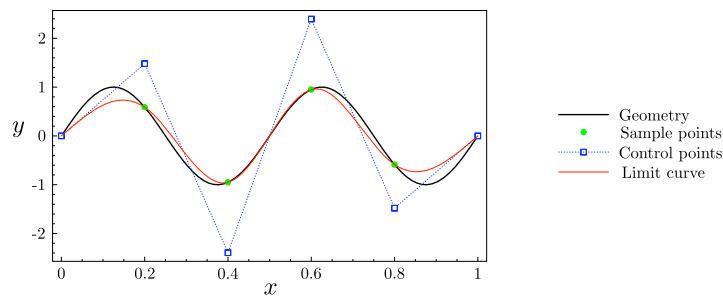
where $\mathcal{P} = \{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_{n_c}\}$ is a set of control points with n_c control points. \mathbf{L} is an evaluation operator of Catmull–Clark curves or surfaces. Set $\xi = (0, 0)$ to ensure the sample points correspond to the control vertices and $n_s \equiv n_c$, then \mathbf{L} is a square matrix. The control points can be calculated as

$$\mathcal{P} = \mathbf{L}^{-1} \mathcal{S}. \quad (18)$$

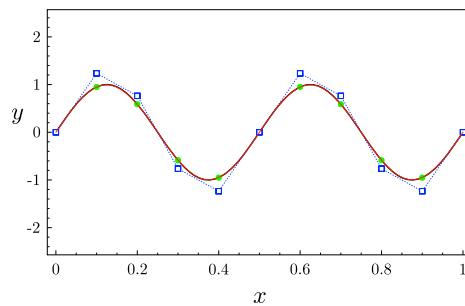
If more sampling points n_s are chosen than the required number of control points n_c , then \mathbf{L} is invertible, a least-squares method is used to obtain a set of control points $\hat{\mathcal{P}}$ that minimises $\|\mathcal{S} - \mathbf{L} \mathcal{P}\|^2$ as

$$\hat{\mathcal{P}} = [\mathbf{L}^T \mathbf{L}]^{-1} \mathbf{L}^T \mathcal{S}. \quad (19)$$

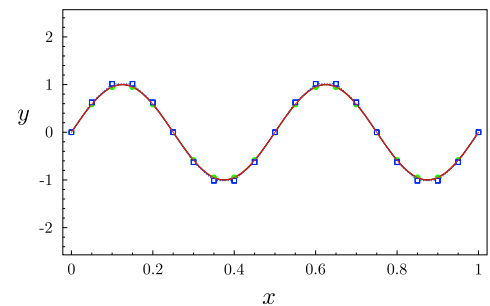
Fig. 5 The process of constructing cubic B-spline curves to approximate a given curve



(a) Fitting curve with $n_s = 6$ sampling points.



(b) Fitting curve with $n_s = 11$ sampling points.



(c) Fitting curve with $n_s = 21$ sampling points.

Figure 5 shows an example of fitting a geometry using cubic B-spline curves based on the Catmull–Clark subdivision algorithm. The given curve is defined as $y = \sin(4\pi x)$. Figure 5a shows that 6 sample points are chosen from the given curve and one assembles the evaluation operator for these sampling points. The control points can be obtained by solving (18). Using these control points, the limit curve can be interpolated. Since 6 sample points is not sufficient to capture the given curve, the limit curve is significantly different to the given curve. Figures 5b and c show the curve fitting with 11 and 21 sample points, respectively. Increasing the number of samples points, the limit curve converges to the given curve.

3.2 Adaptive quadrature rule for element with an extraordinary vertex

In numerical analysis, a Gauss quadrature rule is applied to integrate over Catmull–Clark subdivision elements. A one dimensional quadrature rule with n_q Gauss points can exactly evaluate the integrals for polynomials of degree up to $2n_q - 1$. The polynomial degree of a cubic B-spline function is 3. Because the basis functions of a Catmull–Clark subdivision element in regular element patch are generated as the tensor product of two cubic splines, 2×2 Gauss points can be used in this case. However, if a Catmull–Clark subdivision element has an extraordinary vertex, the basis functions are generated by Eq. (12). In this case, basis functions are not polynomials and the derivatives of the basis functions suffer from the

singular parametrisation problem, see Remark 1. Thus, the standard Gauss quadrature can not be used to evaluate the element integral. Inspired by [35], an adaptive quadrature rule, well suited to Catmull–Clark subdivision surfaces is adopted by integration at a number of levels of subdivided elements. With n_d levels of subdivisions, the element is subdivided into $3n_d + 1$ sub-elements as shown in Fig. 4d. The sub-elements can be evaluated using cubic B-splines with new control vertices except for the ones having an extraordinary vertex. Thus the Gauss quadrature rule can be used to evaluate the integrals in $3n_d$ sub-elements. With a number of subdivisions, the integration error can be reduced. In this work, $n_d = 7$ is chosen in order to obtain sufficiently accurate values of the integrals.

3.3 Penalty method for applying boundary condition

The basis functions do not have the Kronecker delta and interpolating properties, so boundary conditions can not be directly applied using conventional methods. The method used here is a penalty method which uses a penalty parameter and boundary mass matrix to apply the boundary conditions approximately. It preserves the symmetry of the system matrix and does not increase its size. However, the penalty parameter should be carefully selected. If fine meshes with more degrees of freedom are adopted, a larger penalty parameter must be chosen. The Dirichlet boundary condition is defined as

$$u = \bar{u} \quad \text{on} \quad \partial \Gamma. \quad (20)$$

An L_2 projection is used for applying the Dirichlet boundary condition, where for test function v , one obtains

$$\int_{\partial \Gamma} v u \, dL = \int_{\partial \Gamma} v \bar{u} \, dL, \quad \forall v \in H_0^1(\partial \Gamma). \quad (21)$$

Using the cubic B-spline functions in Eq. (2) to discretise u and v and the same strategy for formulating the system matrix, one introduces a boundary mass matrix as

$$\mathbf{M}_b = \mathbf{A} \sum_{e=1}^{n_b} \mathbf{G}^e(\xi^i) |J^j(\mathbf{x}(\xi^i))| w^i, \quad (22)$$

where n_b is the number of boundary elements, and

$$\mathbf{G}^e(\xi) = \begin{bmatrix} N_1(\xi)N_1(\xi) & N_1(\xi)N_2(\xi) & \cdots & N_1(\xi)N_{n_b}(\xi) \\ N_2(\xi)N_1(\xi) & N_2(\xi)N_2(\xi) & \cdots & N_2(\xi)N_{n_b}(\xi) \\ \vdots & \vdots & \ddots & \vdots \\ N_{n_b}(\xi)N_1(\xi) & N_{n_b}(\xi)N_2(\xi) & \cdots & N_{n_b}(\xi)N_{n_b}(\xi) \end{bmatrix}. \quad (23)$$

The right hand side vector for applying the boundary conditions is thus

$$\mathbf{f}_b = \mathbf{A} \sum_{j=1}^{n_b} \bar{u}(\mathbf{x}(\xi_j)) \mathbf{N}^j(\xi_j) |J^j(\mathbf{x}(\xi_j))| w_j. \quad (24)$$

Then the discrete system of equations arising from (21) is

$$\mathbf{M}_b \mathbf{u} = \mathbf{f}_b. \quad (25)$$

We note that the elements for applying boundary conditions are the discretisation of the surface boundary which are one dimensional cubic B-spline curves and only one-dimensional Gauss quadrature rule is used for integration. However, one uses the global degrees of freedom indices to assemble \mathbf{M}_b and \mathbf{f}_b , so that they have the same size as the system matrix and global right-hand side vector, respectively. Assume the system of equations is expressed as $\mathbf{K} \mathbf{u} = \mathbf{f}$, where \mathbf{K} is the system matrix, \mathbf{u} is the global coefficients vector to be solved for, and \mathbf{f} is global right-hand side vector. Then, we scale \mathbf{M}_b and \mathbf{f}_b using a penalty factor β and combine them with the systems of equations as

$$[\mathbf{K} + \beta \mathbf{M}_b] \mathbf{u} = \mathbf{f} + \beta \mathbf{f}_b. \quad (26)$$

The Dirichlet boundary condition (20) is here weakly applied to the system of equations. A relatively large penalty factor $\beta = 10^8$ is selected for all numerical examples. It is sufficiently large to ensure good satisfaction of the constraint but

not too large so as to significantly impact the conditioning of the system.

4 Laplace–Beltrami problem

The governing partial differential equation which we want to solve to illustrate fundamental features of subdivision surfaces is given by

$$-\Delta_\Gamma u = f \quad \text{on} \quad \Gamma, \quad (27)$$

where Γ is a two dimensional manifold (with outward unit normal vector \mathbf{n}) in three dimensional space \mathbb{R}^3 and $\Delta_\Gamma(\bullet)$ is the Laplace–Beltrami operator (also called surface Laplacian operator). The Dirichlet boundary condition is expressed in (20). We will use a manufactured solution to compute against the approximate solution. The Laplace–Beltrami operator is defined by

$$\Delta_\Gamma(\bullet) = \nabla_\Gamma \cdot \nabla_\Gamma(\bullet), \quad (28)$$

where $\nabla_\Gamma(\bullet)$ is the surface gradient operator defined by

$$\nabla_\Gamma(\bullet) = [\mathbf{I} - \mathbf{n} \otimes \mathbf{n}] \cdot \nabla(\bullet). \quad (29)$$

Hence the surface gradient of a scalar function v can be calculated as the spatial gradient subtracted by its normal part as

$$\nabla_\Gamma v = \nabla v - \mathbf{n}[\mathbf{n} \cdot \nabla v], \quad (30)$$

where $\nabla(\bullet)$ is the spatial gradient operator. Hence the surface Laplacian of v is given by

$$\Delta_\Gamma v = \Delta v - \mathbf{n} \cdot [\nabla^2 v \cdot \mathbf{n}] - [\mathbf{n} \cdot \nabla v] [\nabla \cdot \mathbf{n} - \mathbf{n} \cdot [\nabla \mathbf{n} \cdot \mathbf{n}]], \quad (31)$$

where $\nabla^2 v$ is the Hessian matrix of v , and $\nabla \mathbf{n}$ is the gradient of the normal vector, which is arranged in a matrix as

$$\nabla \mathbf{n} = \begin{bmatrix} \frac{\partial n_1}{\partial x_1} & \frac{\partial n_2}{\partial x_1} & \frac{\partial n_3}{\partial x_1} \\ \frac{\partial n_1}{\partial x_2} & \frac{\partial n_2}{\partial x_2} & \frac{\partial n_3}{\partial x_2} \\ \frac{\partial n_1}{\partial x_3} & \frac{\partial n_2}{\partial x_3} & \frac{\partial n_3}{\partial x_3} \end{bmatrix}. \quad (32)$$

We define the total curvature at a surface point $\mathbf{x} \in \Gamma$ as the surface divergence of the normal, that is $c(\mathbf{x}) := \nabla_\Gamma \cdot \mathbf{n}$. For a given manufactured solution u^m , the right hand side of Eq. (27) can thus be computed as

$$f(\mathbf{x}) = -\Delta u^m(\mathbf{x}) + \mathbf{n}(\mathbf{x}) \cdot [\nabla^2 u^m(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})] + c(\mathbf{x})[\mathbf{n}(\mathbf{x}) \cdot \nabla u^m(\mathbf{x})], \quad \mathbf{x} \in \Gamma. \quad (33)$$

5 Galerkin formulation

The weak formulation of problem (27) is

$$\int_{\Gamma} \nabla_{\Gamma} u \cdot \nabla_{\Gamma} v \, d\Gamma = \int_{\Gamma} f v \, d\Gamma, \quad \forall v \in H_0^1(\Gamma), \quad (34)$$

where v is an admissible test function. The weak formulation is partitioned into n_e number of elements, as

$$\sum_{k=1}^{n_e} \int_{\Gamma_e} \nabla_{\Gamma} u \cdot \nabla_{\Gamma} v \, d\Gamma = \sum_{k=1}^{n_e} \int_{\Gamma_e} f v \, d\Gamma. \quad (35)$$

Discretising v , ∇u and ∇v using the Catmull–Clark basis functions \mathbf{N} given in Eq. (7) produces

$$\nabla_{\Gamma} \hat{N}_A = \frac{\partial \hat{N}_A}{\partial \bar{\xi}} \cdot \frac{\partial \bar{\xi}}{\partial \xi} \cdot \mathbf{J}^{-1} \quad (38)$$

and

$$\mathbf{J} = \frac{\partial \mathbf{x}}{\partial \bar{\xi}} \cdot \frac{\partial \bar{\xi}}{\partial \xi} = \sum_{A=0}^{2\kappa+7} \frac{\partial \hat{N}_A}{\partial \bar{\xi}} \mathbf{P}_A \cdot \frac{\partial \bar{\xi}}{\partial \xi}. \quad (39)$$

Integrating the discrete problem using Gauss quadrature, the system of Eq. 34 becomes

$$\underbrace{\left[\mathbf{A} \sum_{e=1}^{n_e} \mathbf{D}^e(\xi^i) |\mathbf{J}(\mathbf{x}(\xi^i))| w^i \right]}_{\mathbf{K}} \mathbf{u} = \underbrace{\mathbf{A} \sum_{e=1}^{n_e} f(\mathbf{x}(\xi^i)) \mathbf{N}^e(\xi^i) |\mathbf{J}(\mathbf{x}(\xi^i))| w^i}_{\mathbf{f}}, \quad (40)$$

where \mathbf{A} is the assembly operator and

$$\mathbf{D}^e(\xi) = \begin{bmatrix} \nabla_{\Gamma} N_1(\xi) \cdot \nabla_{\Gamma} N_1(\xi) & \nabla_{\Gamma} N_1(\xi) \cdot \nabla_{\Gamma} N_2(\xi) & \cdots & \nabla_{\Gamma} N_1(\xi) \cdot \nabla_{\Gamma} N_{n_b}(\xi) \\ \nabla_{\Gamma} N_2(\xi) \cdot \nabla_{\Gamma} N_1(\xi) & \nabla_{\Gamma} N_2(\xi) \cdot \nabla_{\Gamma} N_2(\xi) & \cdots & \nabla_{\Gamma} N_2(\xi) \cdot \nabla_{\Gamma} N_{n_b}(\xi) \\ \vdots & \vdots & \ddots & \vdots \\ \nabla_{\Gamma} N_{n_b}(\xi) \cdot \nabla_{\Gamma} N_1(\xi) & \nabla_{\Gamma} N_{n_b}(\xi) \cdot \nabla_{\Gamma} N_2(\xi) & \cdots & \nabla_{\Gamma} N_{n_b}(\xi) \cdot \nabla_{\Gamma} N_{n_b}(\xi) \end{bmatrix}. \quad (41)$$

$$\begin{aligned} v &= \sum_{A=1}^{n_b} N_A v_A, \\ \nabla_{\Gamma} u &= \sum_{A=1}^{n_b} \nabla_{\Gamma} N_A u_A = \sum_{A=1}^{n_b} \frac{\partial N_A}{\partial \bar{\xi}} \cdot \mathbf{J}^{-1} u_A, \\ \nabla_{\Gamma} v &= \sum_{A=1}^{n_b} \nabla_{\Gamma} N_A v_A = \sum_{A=1}^{n_b} \frac{\partial N_A}{\partial \bar{\xi}} \cdot \mathbf{J}^{-1} v_A, \end{aligned} \quad (36)$$

where \mathbf{J} is the surface Jacobian for the manifold, given in a matrix form as

$$\mathbf{J} = \frac{\partial \mathbf{x}}{\partial \xi} = \begin{bmatrix} \frac{\partial x_1}{\partial \xi} & \frac{\partial x_1}{\partial \eta} \\ \frac{\partial x_2}{\partial \xi} & \frac{\partial x_2}{\partial \eta} \\ \frac{\partial x_3}{\partial \xi} & \frac{\partial x_3}{\partial \eta} \end{bmatrix}. \quad (37)$$

For details on the computation of \mathbf{J}^{-1} see [46] and for a discussion of superficial tensors such as \mathbf{J} in the context of Laplace–Beltrami equation, see [31]. If the element contains an extraordinary vertex, the shape functions N_A are replaced by \hat{N}_A in Eq. (12). The surface gradient of the shape functions is computed as

n_q is the number of quadrature points in each element, w_i is the weight for i^{th} quadrature point, n_e is the number of elements and n_b is the number of basis functions of the element. The basis functions N^e are replaced by \hat{N}^e if the element e contains an extraordinary vertex. In this case, the basis functions are not differentiable and their derivatives approach positive infinity when points are close to the extraordinary vertex (see Remark 1). Thus $|\mathbf{J}|$ approaches positive infinity at extraordinary vertices. Errors result if quadrature is adopted to integrate the contributions from element containing extraordinary vertices.

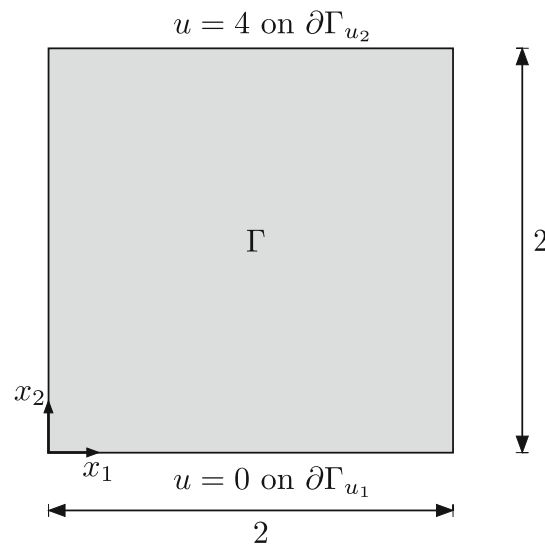
The discrete system of equations to solve is thus given by

$$\mathbf{K} \mathbf{u} = \mathbf{f}. \quad (42)$$

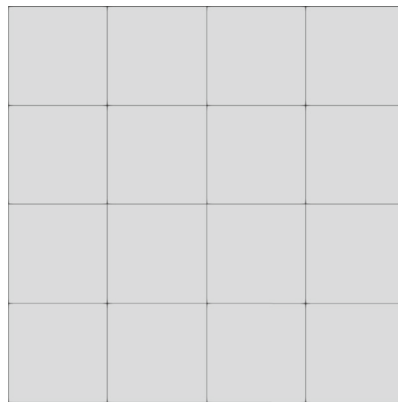
6 Numerical results

A ‘patch test’ [58] on a two-dimensional plate is first presented to assess the consistency and stability of the proposed formulation in a simplified setting. Then, the Laplace–Beltrami equation is solved on both cylindrical and hemispherical surfaces. Convergence studies are conducted. The influence of extraordinary vertices is also investigated.

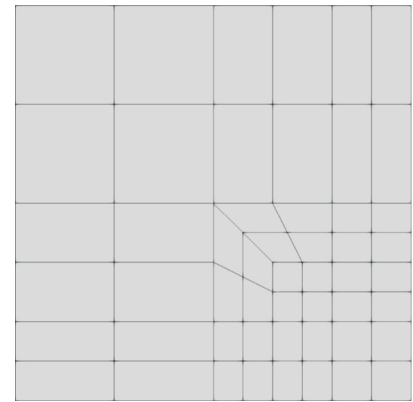
Fig. 6 Schematic of the patch test on a plate



(a) Plate problem setup.



(b) A regular mesh (Mesh 1).



(c) A mesh with extraordinary vertices (Mesh 2).

All numerical results are computed using an open source finite element library: deal.II [2,8].

6.1 ‘Patch test’

The ‘patch test’ is performed on a two dimensional flat plate where the Laplace–Beltrami operator reduces to the Laplace operator. The problem proposed in Sect. 4 reduces to the Poisson problem expressed given by

$$-\Delta u = f \quad \text{on } \Gamma \in \mathbb{R}^2. \quad (43)$$

This partial differential equation is solved on the square plate shown in Fig. 6a with the essential boundary conditions

$$\begin{cases} u = 0 & \text{for } x_2 = 0 \quad (\partial\Gamma_{u_1}) \\ u = 4 & \text{for } x_2 = 2 \quad (\partial\Gamma_{u_2}) \end{cases}. \quad (44)$$

The essential boundary conditions are imposed using the penalty method. Natural homogeneous boundary conditions are applied on the remaining two edges of the plate. Four different manufactured functions for f are used. The functions, analytical solutions for u and their gradients $\frac{\partial u}{\partial x_2}$ are given in Table 1. We investigate both a regular and an irregular mesh. The regular mesh is a 4×4 element patch without extraordinary vertices as shown in Fig. 6b. In all of the tests, a geometry error is absent.

For Test 1, the right hand side $f = 0$ so that $\frac{\partial u}{\partial x_2} = 2$. Solving the equation using the proposed Catmull–Clark subdivision method, the numerical result u_h is exactly 2 everywhere as shown in Fig. 7b. Thus passes the consis-

Table 1 Four test case functions for the plate problem

Test no.	f	u	$\frac{\partial u}{\partial x_2}$
1	0	$2x_2$	2
2	1	$-\frac{1}{2}x_2^2 + 3x_2$	$-x_2 + 3$
3	x_2	$-\frac{1}{6}x_2^3 + \frac{8}{3}x_2$	$-\frac{1}{2}x_2^2 + \frac{8}{3}$
4	$\pi \sin(\pi x_2)$	$\frac{1}{\pi} \sin(\pi x_2) + 2x_2$	$\cos(\pi x_2) + 2$

Test 1 has no right-hand side term, thus the analytical solution u is linear and its gradient is a constant. The analytical solutions for Tests 2 and 3 are quadratic and cubic, respectively, and their gradients are linear and quadratic, respectively. Test 4 has a sine function as the right-hand side term which gives a cosine function as the gradient of the analytical solution

tency test and the eigenvalue of the system matrix are all positive and non-zero after application of the essential boundary conditions. The gradient $\frac{\partial u}{\partial x_2}$ for Test 2 and 3 are linear and quadratic respectively. Recall that when interpolating functions in elements with edges on physical boundaries, the basis functions are modified, see Eqs. (3) and (4). In other words, the gradients of the function u are expected to be constant at boundaries. Figure 7a, c and e show the numerical results for these tests. The results are smooth and capture the analytical solutions well. Figure 7d and f compare the numerical results of $\frac{\partial u}{\partial x_2}$ to the analytical solution for Tests 2 and 3. The Catmull–Clark subdivision method is also compared to linear and quadratic Lagrangian finite element methods. There is a substantial error in both boundary regions in Test 2 for Catmull–Clark subdivision method. This is because the method imposes the gradient to be constant at both boundaries. The numerical result of the Catmull–Clark subdivision method in Test 3 has a substantial error in the region close to the top boundary ($x_2 = 2$) but captures the gradient in the region close to the bottom boundary ($x_2 = 0$) well because the analytic solution for the gradient in the bottom boundary region is near-constant. These errors at the boundaries will pollute the numerical result in the interior of the domain, which will reduce the convergence rate. The gradients approximated by the linear and quadratic Lagrangian finite elements are piecewise constant and piecewise linear, respectively. The results of the Catmull–Clark subdivision methods for these two tests lies between the linear and quadratic Lagrangian elements. The gradient $\frac{\partial u}{\partial x_2}$ in Test 4 is a cosine function which is non-polynomial and it behaves as a constant in both boundary regions shown in Fig. 7h. The Lagrangian elements only possess C^0 continuity across elements and their gradients hence have jumps between elements. The Catmull–Clark subdivision elements capture the gradients of the given function better as they are C^1 smooth.

Figure 8 shows the plots of normalised global L_2 and H^1 errors against the element size. The normalised global L_2 error is defined by

$$e_{L_2} = \frac{\|u - u_h\|_{L_2}}{\|u\|_{L_2}}, \quad (45)$$

where $\|\bullet\|_{L_2}$ is the L_2 norm defined as $\|\bullet\|_{L_2} = \sqrt{\int_{\Gamma} |\bullet|^2 d\Gamma}$. The normalised global H^1 error is computed as

$$e_{H^1} = \frac{\|u - u_h\|_{H^1}}{\|u\|_{H^1}}, \quad (46)$$

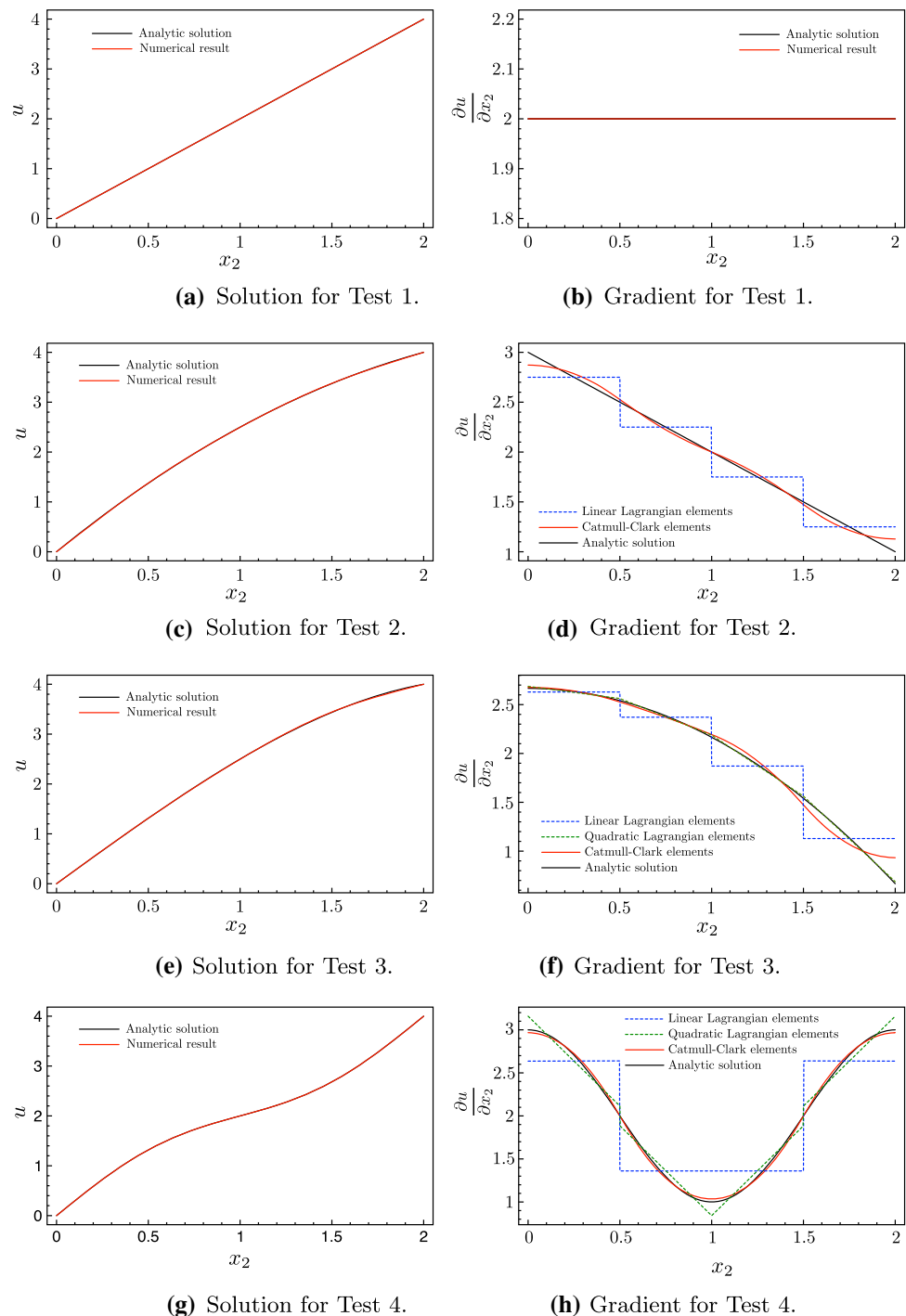
where $\|\bullet\|_{H^1}$ is the H^1 norm defined as $\|\bullet\|_{H^1} = \sqrt{\int_{\Gamma} |\bullet|^2 d\Gamma + \int_{\Gamma} |\nabla(\bullet)|^2 d\Gamma}$. We set the element size of the coarsest mesh as 1. Then, the normalised element size for the refined meshes are $\frac{1}{2}, \frac{1}{4}, \dots$. The convergence rate of Tests 2 and 3 are sub-optimal at 2.5 (L_2 error) and 1.5 (H^1 error). The optimal convergence rate for cubic elements should be $p + 1 = 4$ (L_2 error) and $p = 3$ (H^1 error), where p is the polynomial degree of the basis functions. The numerical result captures the analytical solution well and the convergence rate for Test 4 is optimal. The same convergence study is now repeated starting from a mesh containing extraordinary vertices as shown in Fig. 6c. Figure 8a and b show the plots of normalised element sizes against the L_2 and H^1 errors, respectively. The same convergence rates are obtained for Tests 2 and 3. However, the convergence rate of Test 4 is also reduced to 2.5 (L_2 error) and 1.5 (H^1 error). Figure 8c and d show the plots of normalised element sizes against L_2 and H^1 errors, respectively, for the mesh with an extraordinary vertex.

The Catmull–Clark subdivision method can pass the patch test when the function gradient is a constant but has difficulties to capture the gradients in boundary regions when they do not behave like a constant. When the gradient behaves like a constant in the boundary regions, the optimal convergence rate can be obtained. If this is not the case, a reduction of the convergence rate is observed. The presence of the extraordinary vertex in the patch also reduces the convergence rate. It is also important to note that the Catmull–Clark subdivision elements have advantages in describing non-polynomial functions since their basis functions are cubic and C^2 continuous.

Comparison with NURBS and Lagrangian elements

We now compare the convergence rate associated with Catmull–Clark elements against conventional Lagrangian elements and NURBS. Bézier extraction [12] is adopted to decompose a NURBS surfaces into C_0 Bézier elements to provide an element structure for the isogeometric Galerkin method. This is a widely-used method for isogeometric analysis using T-splines [48]. As the Lagrangian and Bézier elements can fully pass the ‘patch test’, they both have no approximation error for Tests 1, 2 and 3. Figure 9 compares their behaviour in approximating non-polynomial solution

Fig. 7 Solution u and the gradient $\frac{\partial u}{\partial x_2}$ plotted along the line $x_1 = 1$ for the plate test. The numerical results are compared to the analytical solutions for tests 2, 3 and 4. Mesh 1 is used for all tests



in Test 4. Mesh 1 is used for all methods. All methods exhibit an optimal convergence rate. Since no geometry error is involved in the ‘patch test’, the Bézier element provides the same performance as the Lagrangian element without the advantages of exact geometry representation. The Catmull–Clark element is slightly more accurate than other two methods for this specific test.

6.2 The Laplace–Beltrami equation

The following sections will solve the Laplace–Beltrami Eq. (27) on different two dimensional Catmull–Clark subdivision manifolds. An analytical solution of function u^m is manufactured as

$$u^m(\mathbf{x}) = \sin(\pi x_1) \cos(\pi x_2) e^{x_3}, \quad (47)$$

Fig. 8 Convergence study for Tests 2, 3 and 4 using the regular mesh (Mesh 1) and the mesh with an extraordinary vertex (Mesh 2)

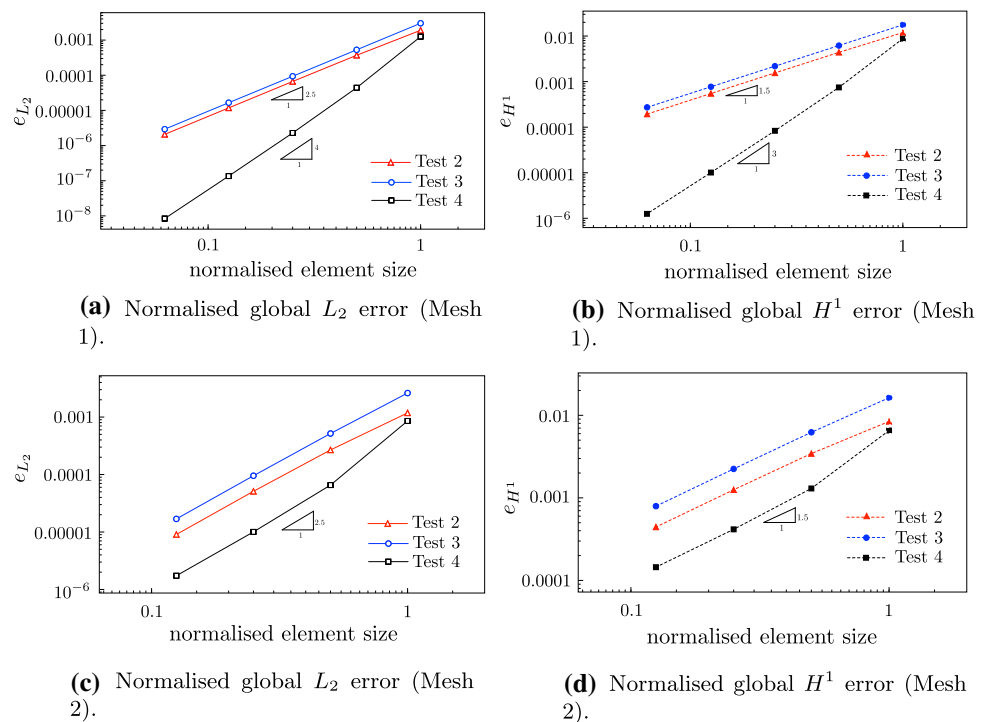
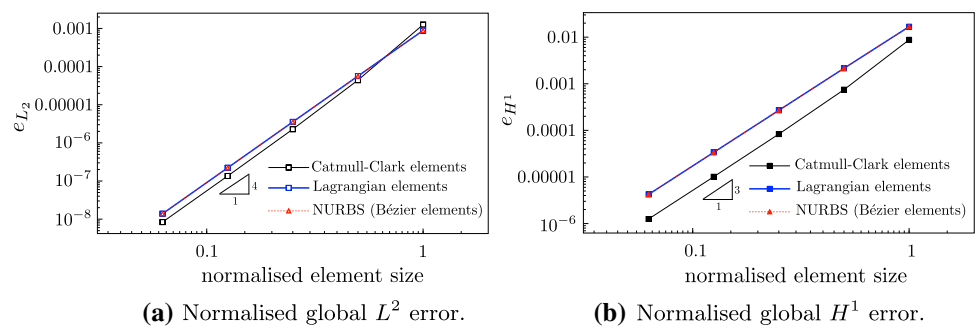


Fig. 9 Convergence studies for Test 4 using Mesh 1. The Catmull–Clark elements are compared with Lagrangian elements and NURBS (Bézier elements). $p = 3$ for all cases



where $\mathbf{x}(x_1, x_2, x_3)$ is a point on the two dimensional manifold in three dimensional space. Applying the Laplacian operator on u^m gives

$$\Delta u^m(\mathbf{x}) = -2\pi^2 \sin(\pi x_1) \cos(\pi x_2) e^{x_3} + \sin(\pi x_1) \cos(\pi x_2) e^{x_3}. \quad (48)$$

Then, the Hessian matrix $\nabla^2 u(\mathbf{x})$ is computed as

$$\nabla^2 u^m(\mathbf{x}) = \begin{bmatrix} -\pi^2 \sin(\pi x_1) \cos(\pi x_2) e^{x_3} & -\pi^2 \cos(\pi x_1) \sin(\pi x_2) e^{x_3} & \pi \cos(\pi x_1) \cos(\pi x_2) e^{x_3} \\ -\pi^2 \cos(\pi x_1) \sin(\pi x_2) e^{x_3} & -\pi^2 \sin(\pi x_1) \cos(\pi x_2) e^{x_3} & -\pi \sin(\pi x_1) \sin(\pi x_2) e^{x_3} \\ \pi \cos(\pi x_1) \cos(\pi x_2) e^{x_3} & -\pi \sin(\pi x_1) \sin(\pi x_2) e^{x_3} & \sin(\pi x_1) \cos(\pi x_2) e^{x_3} \end{bmatrix} \quad (49)$$

The second term in (33) includes the normal vector $\mathbf{n}(\mathbf{x})$ and its gradient which can not be computed analytically. In the present work, an L^2 projection is used to compute the coefficients of normal vectors associated with control points,

denoted by $\hat{\mathbf{n}}$, in order to numerically interpolate the normal vector derivatives at any surface points, thus

$$\frac{\partial \mathbf{n}}{\partial \mathbf{x}} = \sum_{A=1}^{n_b} \frac{\partial N_A}{\partial \mathbf{x}} \hat{\mathbf{n}}_A. \quad (50)$$

6.2.1 Cylindrical surface example

The first numerical example considered is a cylindrical surface. The analysis domain of the problem is the cylindrical

Fig. 10 The geometry is given in (a) and (b) is the control mesh which constructs the best approximating Catmull–Clark subdivision surface of the given geometry. The control mesh is generated using least-squares fitting. c Shows the numerical result u_h on the cylindrical surface

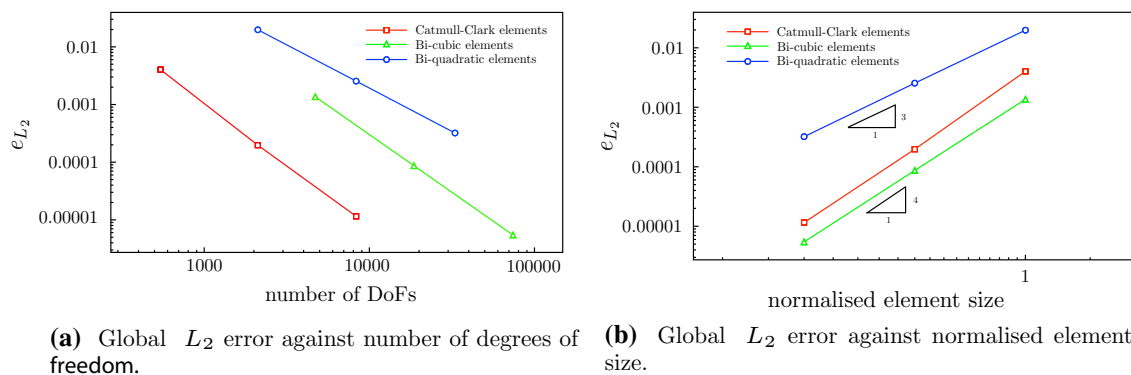
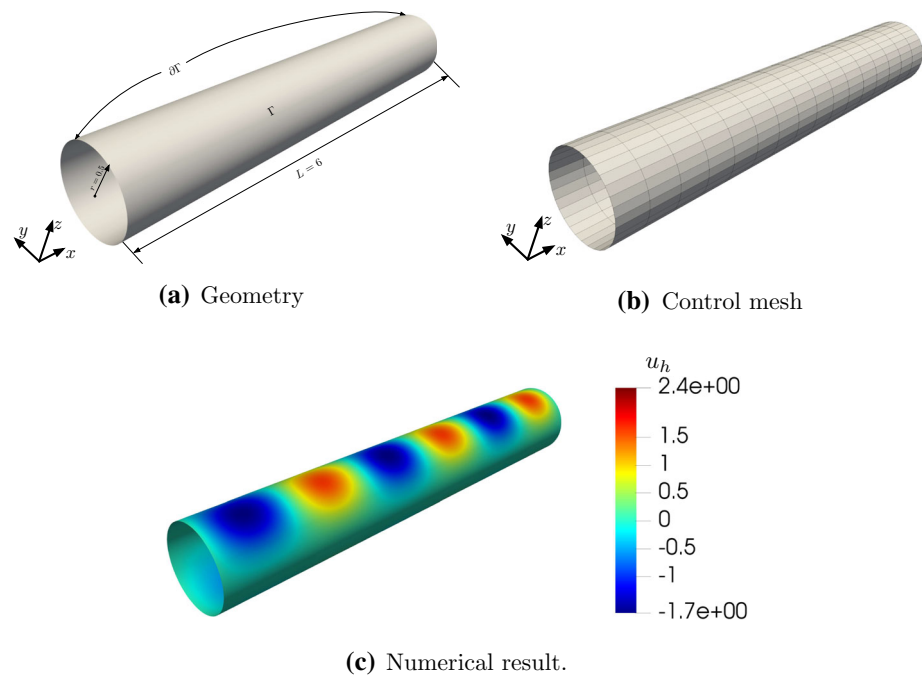


Fig. 11 Convergence study for cylindrical example: comparison of the Catmull–Clark elements to the quadratic and cubic Lagrangian elements

surface shown in Fig. 10a. Surfaces fitting methods are used to construct the control mesh, see Sect. 3.1. The first level control mesh is shown in Fig. 10b. This has no extraordinary vertices. The Laplace–Beltrami problem on this manifold domain is solved using the Galerkin formulation presented in Sect. 5. Essential boundary conditions are applied on $\partial\Gamma$. The right-hand side function f is computed using the definition in Eq. (33). Figure 10c shows the numerical result u_h which matches the manufactured analytical solution (47) very well.

A convergence study is now conducted for this geometry. The refined control meshes are constructed using the least-squares fitting method described in Sect. 3.1. Figure 11 compares the convergence rates between Catmull–Clark subdivision surfaces with two different order Lagrangian elements. In this example, the shortcoming caused by extraordinary vertices and boundary gradients are not present, and

the Catmull–Clark subdivision surfaces have the same convergence rate $p + 1$ as cubic Lagrangian elements.

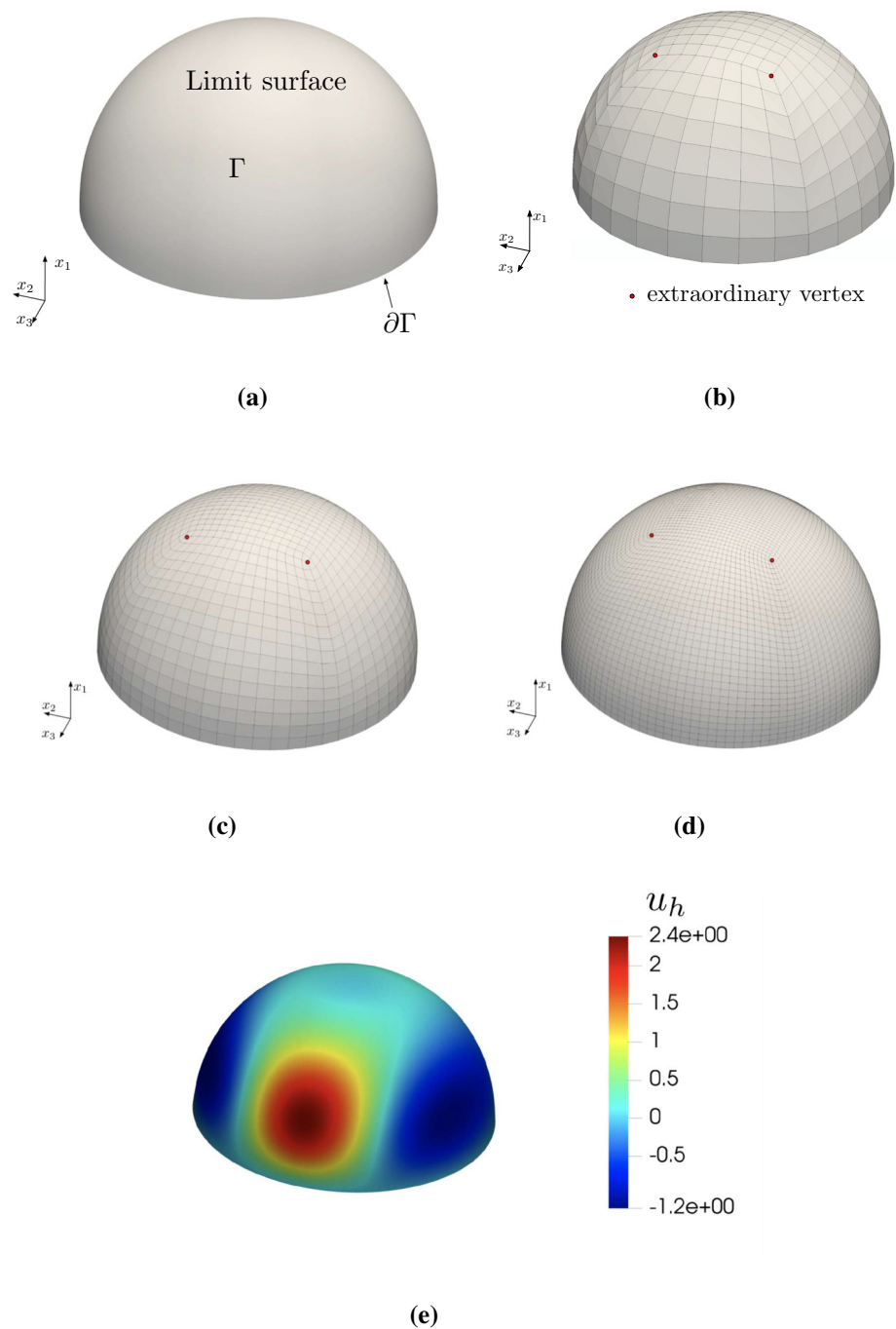
6.2.2 Hemispherical surface example

The second geometry investigated is a hemispherical surface with radius equal to 1 as shown in Fig. 12a. We use the same strategy to fit the Catmull–Clark subdivision surfaces to the hemispherical surface. The control mesh shown in Fig. 12b is generated to discretise the surface into a number of Catmull–Clark elements. The control mesh has four extraordinary vertices. Figure 12e shows the solution u_h .

Convergence study with an isogeometric approach

In engineering, designers usually do not know the geometry of the product in advance. The geometry information is purely from the CAD model. Catmull–Clark subdivision surfaces, as a design tool, provide the geometry which is the

Fig. 12 **a** is a hemispherical surface. **b** is the control mesh for constructing subdivision surfaces to fit the hemispherical surface. **c** is 1-Level refined mesh for the hemispherical surface. **d** is 2-level refined mesh for the hemispherical surface. **e** shows the numerical result u_h on this surface



design of the engineering product. In this case, engineers do not need to approximate the given geometry with Catmull–Clark elements. They can directly adopt the discretisation from the CAD model for analysis. For example, we adopt the control mesh shown in Fig. 12b as the initial control mesh. It can be used to generate a limit surface approximating a hemisphere, as shown in Fig. 12a, with Catmull–Clark subdivision bases. It is important to note the limit surface is not an exact hemisphere since it is evaluated using cubic basis spline functions. However, this surface is the domain of

our problem and it will stay exact the same during the entire analysis (isogeometric) and h -refinement with subdivision algorithm will not change the geometry.

The same problem is solved on the subdivision surfaces. A convergence study is done with another two levels of subdivision control mesh as shown in Fig. 12c and d. Note, refinement does not change the number of extraordinary vertices. The two new meshes still have four extraordinary vertices. The two control meshes can be used to evaluate the same limit surface shown in Fig. 12a. The Catmull–Clark

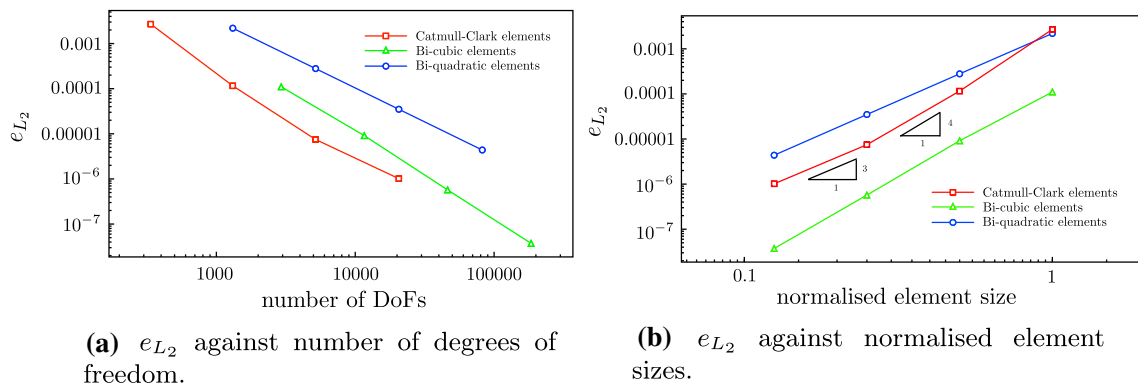
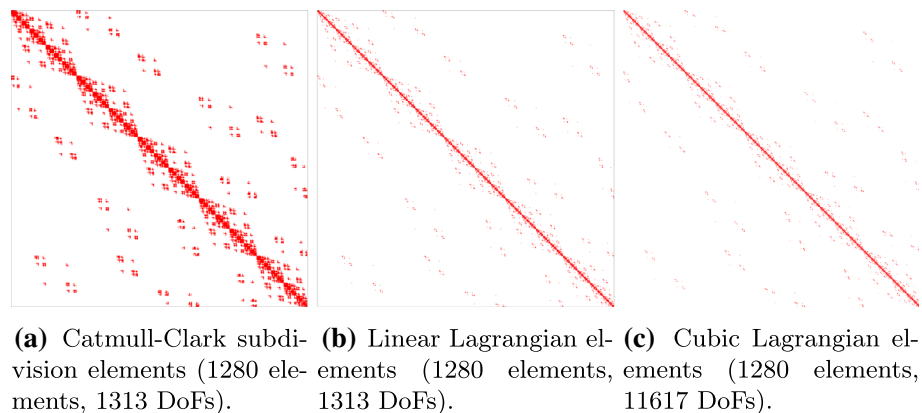


Fig. 13 Convergence study: comparison of the Catmull–Clark elements with the quadratic and cubic Lagrangian elements

Fig. 14 Comparison of sparsity patterns between the Catmull–Clark elements and the Lagrangian elements



subdivision surfaces are compared with quadratic and cubic Lagrangian elements. Generally, Catmull–Clark subdivision elements can achieve higher accuracy per degree of freedom than Lagrangian elements. From the initial to the second level of mesh refinement, the Catmull–Clark subdivision elements have a similar convergence rate to cubic Lagrangian elements. After that, the convergence rate is equivalent to quadratic Lagrangian elements as shown in Fig. 13.

Sparsity patterns

Figure 14a shows the sparsity pattern of the system matrix \mathbf{K} for the Catmull–Clark subdivision discretisation. The size of the matrix is the same as the system matrix assembled using a linear Lagrange discretisation. However, because the Catmull–Clark subdivision discretisation uses cubic basis functions with non-local support and there are 16 shape functions in a subdivision element with no extraordinary vertex, the number of non-zero entries in columns and rows is more than the linear Lagrange discretisation (i.e. the sparsity is decreased and the bandwidth increased). Thus, the system matrix of a Catmull–Clark subdivision discretisation has the same size but is denser than the linear Lagrange discretisation shown in 14b. Figure 14c is the sparsity patterns of cubic Lagrange discretisations. p -refinement increase the number of degrees of freedom as well as the number of

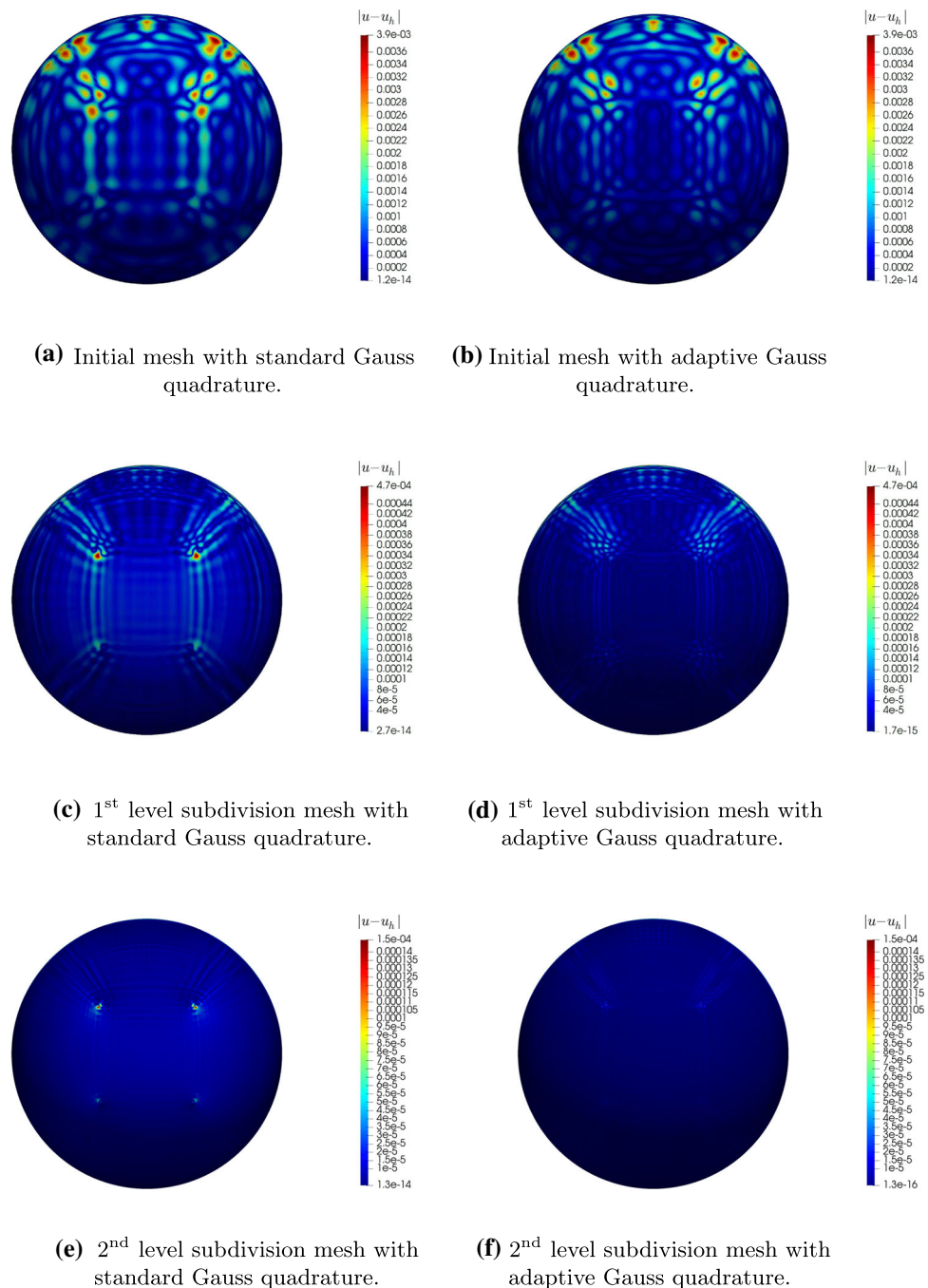
non-zero entries in rows and columns. Thus there is no significant change in the density of the system matrices. The Catmull–Clark subdivision discretisation has the same number of non-zero entries in each row or column as the cubic Lagrangian discretisation but has a much smaller size.

6.3 Investigation of extraordinary vertices

Quadrature error

The presence of extraordinary vertices leads to difficulties in integration as described in Sect. 3.2. Figure 15a, c and e show the point-wise errors at surface points for three levels of mesh refinement using the standard Gauss quadrature rule. The number of extraordinary vertices remains 4 after refinement. For the analysis using the initial mesh, the error in the regions around extraordinary vertices have similar magnitudes to the other regions. However, after a level of refinement, the error in the other regions is reduced more than the area around the four extraordinary vertices. After the second refinement, the error is concentrated in the areas around the four extraordinary vertices. Figure 15b, d and f plot the point-wise errors on the same mesh analysed with the adaptive quadrature rule shown in Sect. 3.2. The errors around extraordinary vertices are now decreased.

Fig. 15 Point-wise error $|u - u_h|$ plots over spherical surfaces

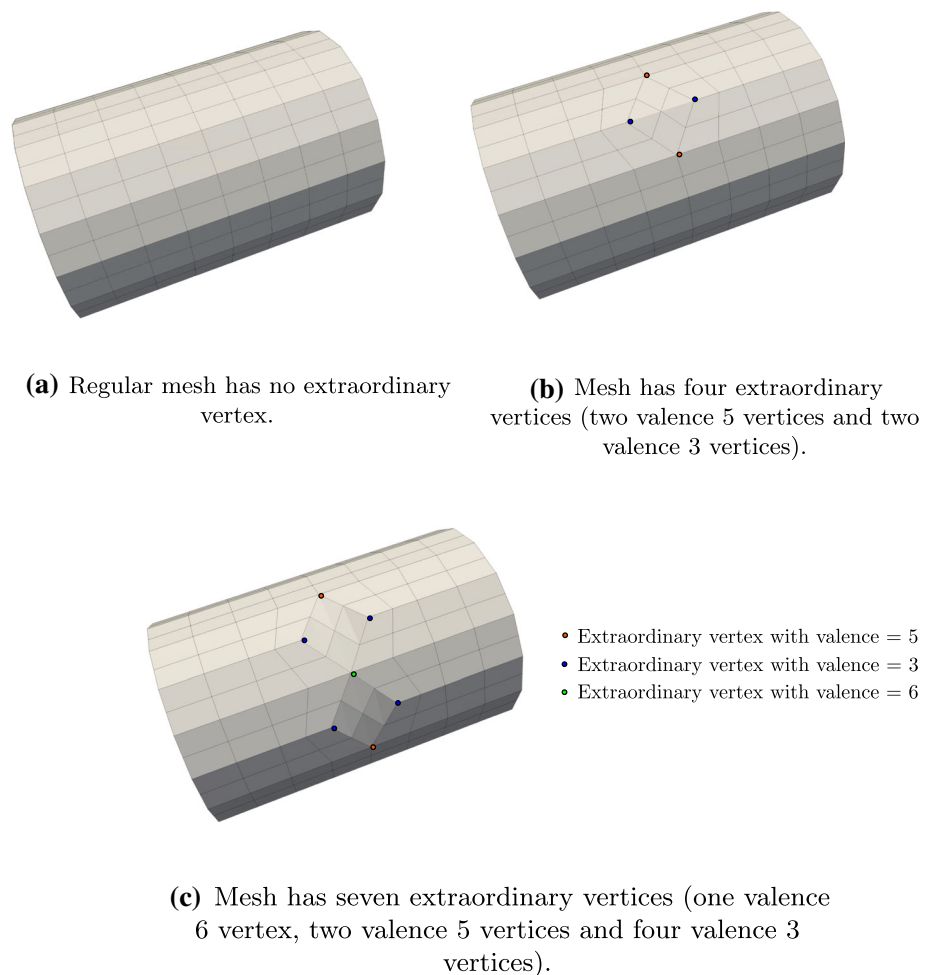


Approximation error

The presence of extraordinary vertices introduces approximation errors. Then we investigate the effect of the number and valence of extraordinary vertices on numerical accuracy. Figure 16a, b and c are three control meshes with different numbers of extraordinary vertices. Figure 16a shows a control mesh without an extraordinary vertex. Figure 16b shows a control mesh with four extraordinary vertices, including two vertices with a valence of 3 and two vertices with a valence of 5. The control mesh in Fig. 16c has seven extraordinary ver-

tices, including four vertices with a valence of 4, two vertices with a valence of 5 and one vertex with a valence of 6. It is important to note the three different control meshes construct different but similar geometries. The Laplace–Beltrami problem is solved using the Galerkin formulation with the same right-hand side function f computed in (33). Both standard and adaptive Gauss quadrature rules are used for all cases. Figure 17a, c and e show the solution of u on the surfaces constructed using the three meshes. Because of the similarity of the geometries and solutions, the three cases are used to inves-

Fig. 16 Three control meshes with different number of extraordinary vertices



tigate the effects of the number of extraordinary vertices in numerical results. The point-wise errors on the three surfaces are shown in Fig. 17b, d and f. Meshes with extraordinary vertices have larger maximum point-wise errors close to the extraordinary vertices, while the mesh without extraordinary vertices has increased uniform point-wise error. Figure 18 shows the convergence rates for the three cases. Meshes without extraordinary vertices can achieve the optimal $p + 1$ convergence rate and $p = 3$. In general, the more extraordinary vertices a mesh contains, the more error results. The extraordinary vertices increase the global errors in the results and reduce the convergence rate. Since the global errors also include quadrature errors, the adaptive quadrature rule serves to reduce the quadrature errors. With the adaptive quadrature rule, the convergence rates are improved for the 4 and 7 extraordinary vertices cases but the results still agree with our assumption that increasing the number and valence of extraordinary vertices will produce higher error.

Computational cost

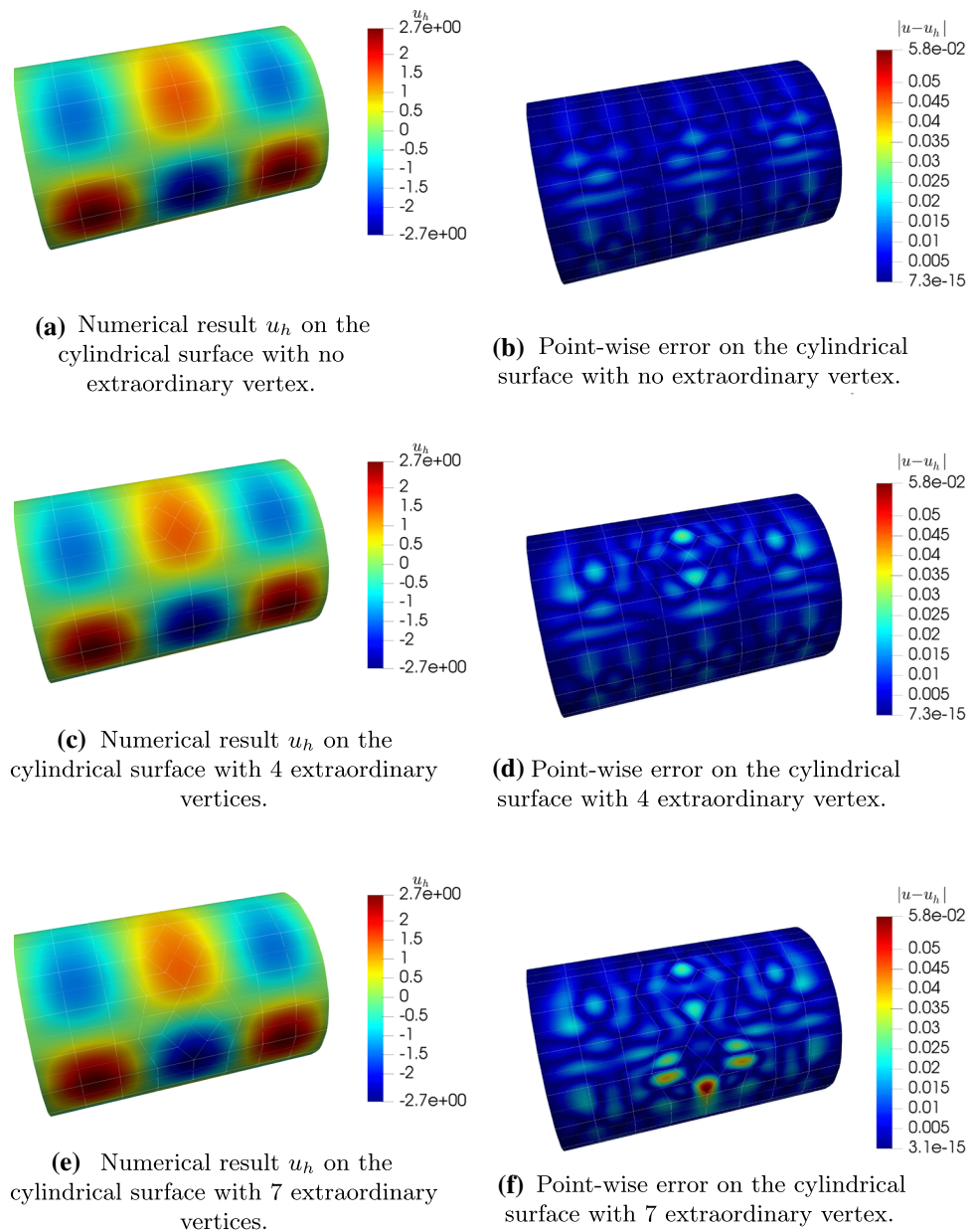
Table 2 compares the computational cost for assembling the system matrix for the standard and adaptive quadrature

rules. Because the number of extraordinary vertices remains constant after subdivision, the difference in computational time between the standard and adaptive quadrature schemes diminishes.

6.4 Complex geometry

This final example considers the ability of the Catmull–Clark method to provide high-order discretisations of complex geometry. The model considered is that of a racing car from CAD and imported into Autodesk Maya [4] for removal of extraneous geometry and the generation of the surface mesh shown in Fig. 19a. Modelling such geometry using NURBS surfaces would require a number of patches to be spliced together. A model based on Catmull–Clark subdivision surface can directly evaluate the smooth limit surface in Fig. 19b using the control mesh containing extraordinary vertices. The minimum bounding box for this model is defined by $[x_i^{\min}, x_i^{\max}]^3 = [-1.047, 1.122] \times [0.097, 0.692] \times [-0.460, 0.460]$. The control mesh has 9154 vertices. The physical domain is naturally discretised into a number of elements expressed as

Fig. 17 Numerical results and point-wise errors on the cylindrical surfaces constructed by three different meshes. The white grids are used to indicate the locations of extraordinary vertices



$$\Gamma = \bigcup_{e=1}^{n_e} \Gamma_e, \quad (51)$$

where $n_e = 9152$ for this example. Figure 19c indicates the domain where essential (Dirichlet) and natural boundary conditions are applied. The essential boundary Γ_d is composed of two parts as

$$\Gamma_d = \Gamma_d^1 + \Gamma_d^1, \quad (52)$$

where

$$\Gamma_d^1 = \{\Gamma_e \mid x_2 < -0.9 \quad \forall \mathbf{x} \in \Gamma_e\}, \quad (53)$$

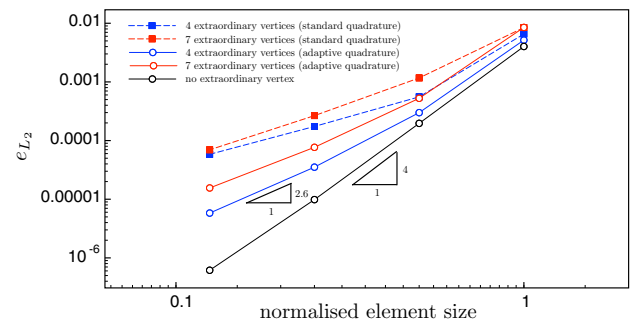
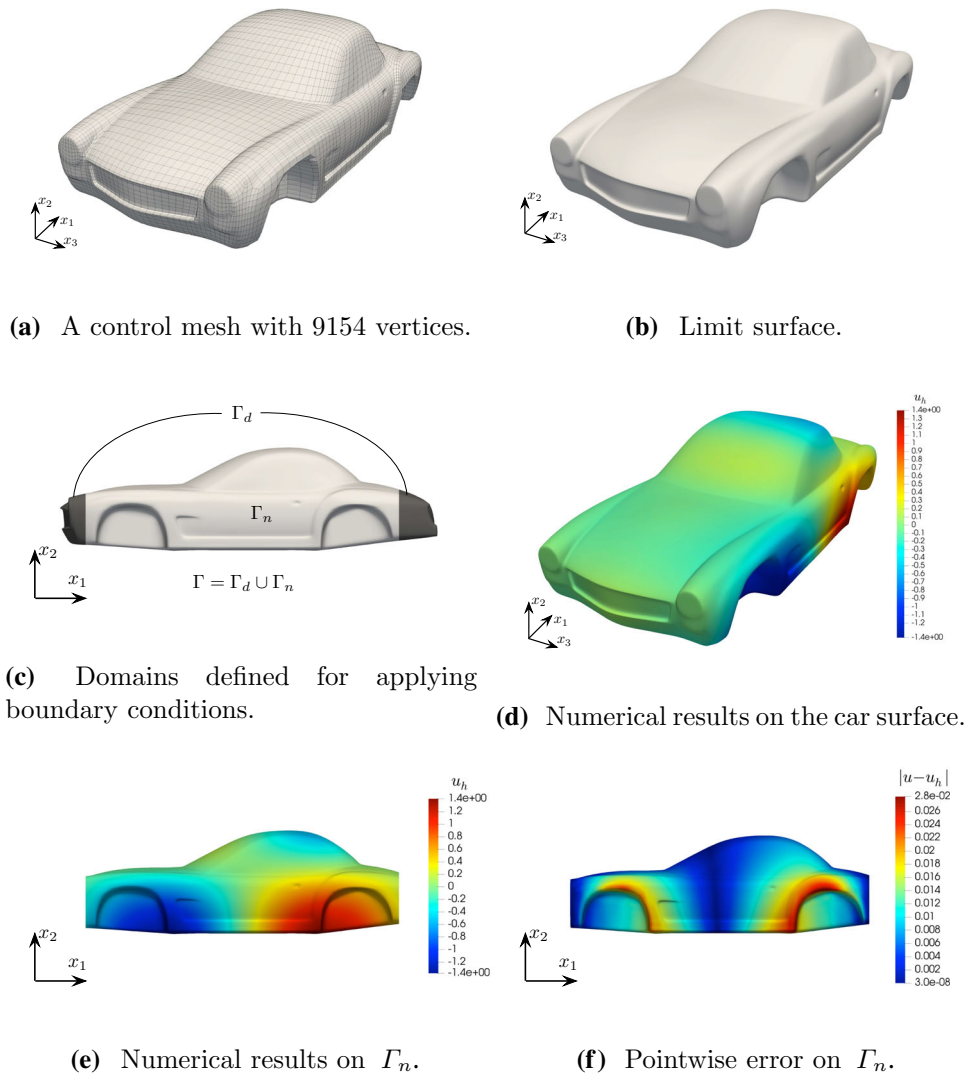


Fig. 18 Convergence study: comparison of the Catmull–Clark subdivisions with different number of extraordinary vertices. The adaptive Gauss quadrature achieves better convergence rates over the standard Gauss quadrature rule

Table 2 The computational times for assembling the system matrix are recorded for tests on cylindrical meshes with different number of extraordinary vertices

Number of extraordinary vertices	n_e	n_d	L_2 error	Assembly time (s)
4	260	0	6.55e−3	2.46
		3	6.71e−3	3.98
		7	6.72e−3	6.39
	1040	0	5.93e−4	9.39
		3	3.48e−4	10.16
		7	3.49e−4	13.87
7	4160	0	1.80e−4	34.9
		3	5.24e−5	35.51
		7	5.31e−5	37.02
	264	7	8.49e−3	9.59
		7	5.54e−4	16.03
		7	8.60e−5	40.24

Standard quadrature ($n_d = 0$) is compared with adaptive quadratures with $n_d = 3$ and 7

Fig. 19 The Laplace–Beltrami problem is solved on a complex car geometry

$$\Gamma_d^2 = \{\Gamma_e \mid x_2 > 0.9 \quad \forall \mathbf{x} \in \Gamma_e\}. \quad (54)$$

The natural boundary conditions is applied to the rest of the domain $\Gamma_n = \Gamma \setminus \Gamma_d$. The numerical result matches the analytical solution well as shown in Fig. 19d. Figure 19e shows the results on Γ_n and a maximum point-wise error 2.8% is observed in Fig. 19f.

7 Conclusions

A thorough study of the isogeometric Galerkin method with Catmull–Clark subdivision surfaces has been presented. The same bases have been used for both geometry and the Galerkin discretisation. The method has been used to solve the Laplace–Beltrami equation on curved two-dimensional manifold embedded in three dimensional space using the Catmull–Clark subdivision surfaces. An approach to fit given geometries using Catmull–Clark subdivision scheme has been outlined. A method to model open boundary geometries without involving ‘ghost’ control vertices, but involving errors in function gradients close to boundary regions, has also been described. The penalty method has been adopted to impose the Dirichlet boundary conditions. The optimal convergence rate of $p + 1$ has been obtained when using a cylindrical control mesh without extraordinary vertices. A reduction of convergence rates has been observed when the function gradients at the boundaries do not behave like constant, or control meshes contain extraordinary vertices. The adaptive quadrature scheme significantly improves the accuracy. The effect of the number and valence of the extraordinary vertices in convergence rates has been investigated and an adaptive quadrature rule implemented. This successfully improved the convergence rates for the proposed method. The convergence rate of the proposed method is not worse than 2.5 (L_2 error) and 1.5 (H^1 error).

In future work, this method will be investigated with problems requiring C^1 continuity such as the deformations of thin shells.

Acknowledgements This work was supported by the UK Engineering and Physical Sciences Research Council grant EP/R008531/1 for the Glasgow Computational Engineering Centre.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copy-

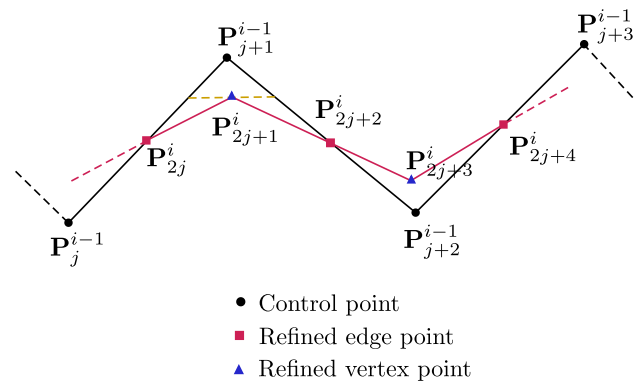


Fig. 20 Computing new control points through the Lane–Riesenfeld algorithm

right holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

A Appendix

A.1 Lane–Riesenfeld subdivision algorithm for curves

The Lane–Riesenfeld algorithm successively refines a curve starting from an initial control polygon. After a number of subdivisions, the curve is limited to a B-spline. Figure 20 illustrates a special case of this subdivision algorithm. The control point \mathbf{P}_{2j}^i in the i^{th} level of refinement is computed from the upper level control points as

$$\mathbf{P}_{2j}^i = \frac{1}{2}\mathbf{P}_j^{i-1} + \frac{1}{2}\mathbf{P}_{j+1}^{i-1}. \quad (55)$$

Point \mathbf{P}_{2j}^i is the mid-point of $\mathbf{P}_j^{i-1} - \mathbf{P}_{j+1}^{i-1}$, and is called an ‘edge point’. The control point \mathbf{P}_{2j+1}^i is computed as

$$\mathbf{P}_{2j+1}^i = \frac{1}{8}\mathbf{P}_j^{i-1} + \frac{3}{4}\mathbf{P}_{j+1}^{i-1} + \frac{1}{8}\mathbf{P}_{j+2}^{i-1}. \quad (56)$$

To compute this point, one needs to connect the mid-points of $\mathbf{P}_{2j}^i - \mathbf{P}_{j+1}^{i-1}$ and $\mathbf{P}_{j+1}^{i-1} - \mathbf{P}_{2j+2}^i$. The point \mathbf{P}_{2j+1}^i is the mid-point of the connecting line. This type of point is called ‘vertex point’. Each ‘vertex point’ is associated with an upper level control point. Figure 21 shows two levels of refinements using the Lane–Riesenfeld algorithm and the limiting result which is a cubic B-spline curve.

A.2 Catmull–Clark subdivision algorithm for surfaces

The application of the subdivision algorithm to surfaces follows in a similar manner to curves. One face in the original

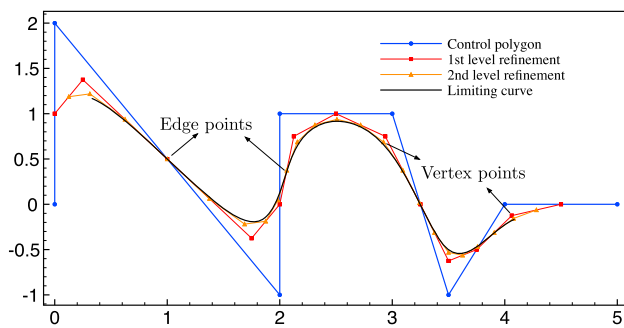


Fig. 21 Schematics of the Lane–Riesenfeld Algorithm

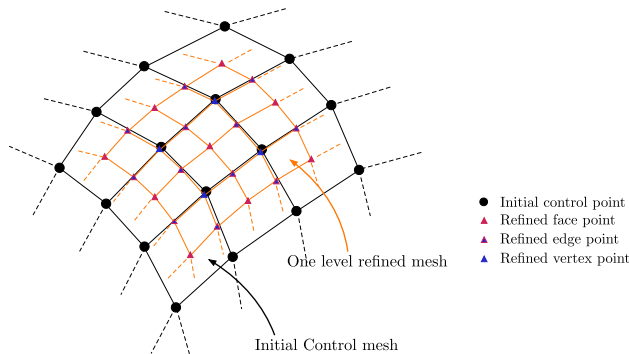


Fig. 22 Catmull–Clark subdivision algorithm for surfaces

control mesh is split into four new faces. For a closed surface, the numbers of faces and control vertices are doubled. Figure 22 shows an example of generating a new control mesh through the Catmull–Clark algorithm.

Similar to the one-dimensional curve, the new refined control points can be classified into three types: face points, edge points and vertex points. The face points in the i^{th} refinement are computed as

$$\mathbf{P}_{2j,2k}^i = \frac{1}{4} \left[\mathbf{P}_{j,k}^{i-1} + \mathbf{P}_{j,k+1}^{i-1} + \mathbf{P}_{j+1,k}^{i-1} + \mathbf{P}_{j+1,k+1}^{i-1} \right], \quad (57)$$

where j and k are indices of control points for orthogonal directions. The face point is the central point of the original face. The edge point is computed as

$$\mathbf{P}_{2j+1,2k}^i = \frac{1}{16} \left[\mathbf{P}_{j,k}^{i-1} + 6\mathbf{P}_{j,k+1}^{i-1} + \mathbf{P}_{j,k+2}^{i-1} + \mathbf{P}_{j+1,k}^{i-1} + 6\mathbf{P}_{j+1,k+1}^{i-1} + \mathbf{P}_{j+1,k+2}^{i-1} \right], \quad (58)$$

or likewise

$$\mathbf{P}_{2j,2k+1}^i = \frac{1}{16} \left[\mathbf{P}_{j,k}^{i-1} + \mathbf{P}_{j,k+1}^{i-1} + 6\mathbf{P}_{j+1,k}^{i-1} + 6\mathbf{P}_{j+1,k+1}^{i-1} + \mathbf{P}_{j+2,k}^{i-1} + \mathbf{P}_{j+2,k+1}^{i-1} \right]. \quad (59)$$

The ‘vertex point’ is computed as

$$\mathbf{P}_{2j+1,2k+1}^i = \frac{1}{64} \left[\mathbf{P}_{j,k}^{i-1} + 6\mathbf{P}_{j,k+1}^{i-1} + \mathbf{P}_{j,k+2}^{i-1} + 6\mathbf{P}_{j+1,k}^{i-1} + 36\mathbf{P}_{j+1,k+1}^{i-1} + 6\mathbf{P}_{j+1,k+2}^{i-1} + \mathbf{P}_{j+2,k}^{i-1} + 6\mathbf{P}_{j+2,k+1}^{i-1} + \mathbf{P}_{j+2,k+2}^{i-1} \right]. \quad (60)$$

Equipped with these formulae, the new control points on the i^{th} level of refinement \mathcal{P}^i can be computed as:

$$\mathcal{P}^i = \mathbf{S} \mathcal{P}^{i-1}, \quad (61)$$

\mathbf{S} is a subdivision operator – a matrix consisting of a set of weights. Each weight is associated with a control point in \mathcal{P}^{i-1} . The weight distributions for different types of control points are shown in Fig. 23. The weight distributions for extraordinary point are shown in Fig. 24. After successive levels of refinements, a smooth B-spline surfaces is obtained.

A.3 Computing control point set for sub-elements

We denote the control points of an irregular patch in Fig. 4a as a set \mathcal{P} . The initial control points of the patch are expressed as

$$\mathcal{P}_0 = \left\{ \mathbf{P}_0^0, \mathbf{P}_1^0, \dots, \mathbf{P}_{2\kappa+6}^0, \mathbf{P}_{2\kappa+7}^0 \right\}. \quad (62)$$

Through one level of subdivision we generate $2\kappa + 17$ new control points (κ is the valence), as shown in Fig. 4b, denoted by

$$\mathcal{P}_1 = \left\{ \mathbf{P}_0^1, \mathbf{P}_1^1, \dots, \mathbf{P}_{2\kappa+15}^1, \mathbf{P}_{2\kappa+16}^1 \right\}. \quad (63)$$

The subdivision step is represented as

$$\mathcal{P}_1 = \mathbf{A} \mathcal{P}_0, \quad (64)$$

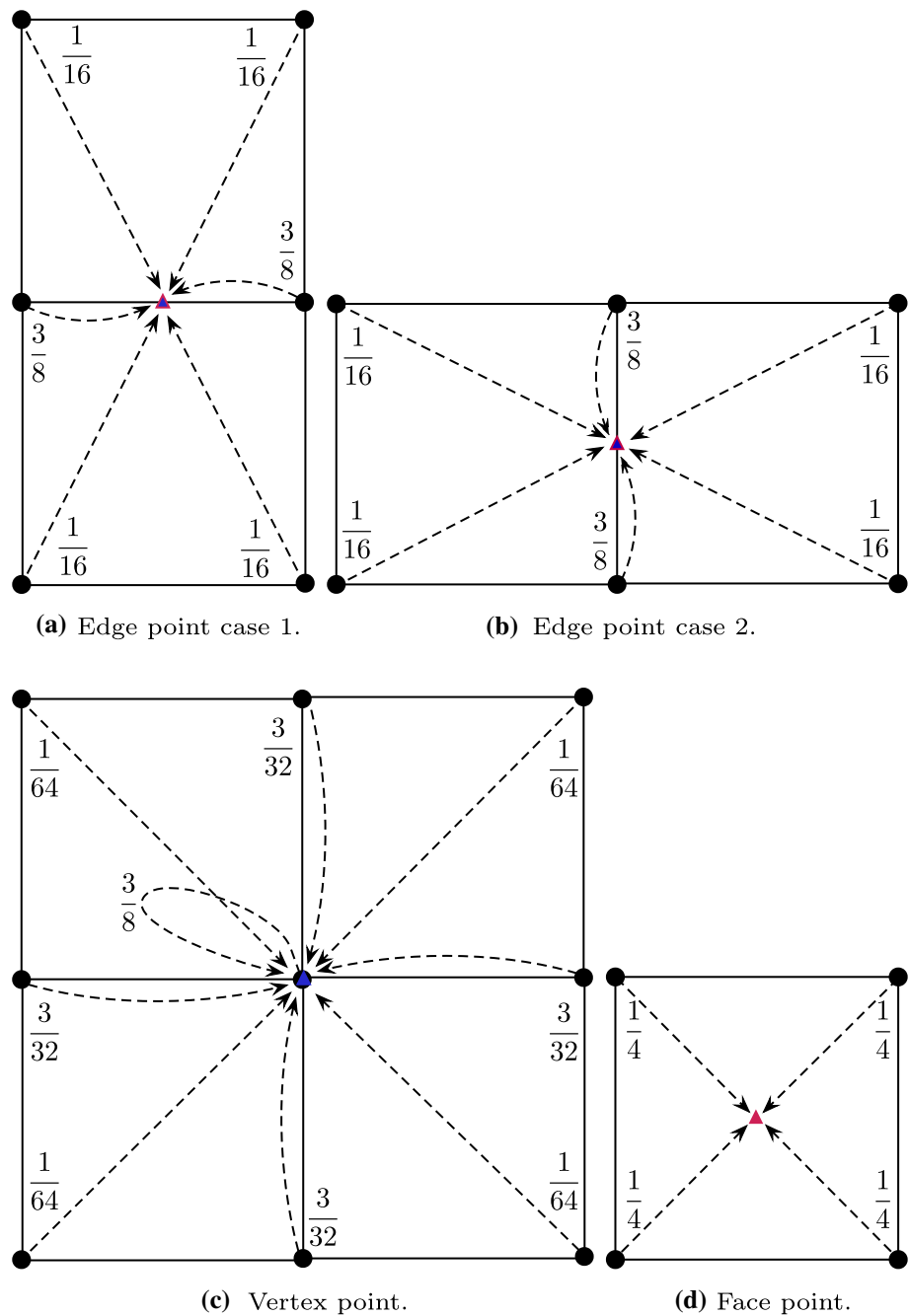
where \mathbf{A} is the subdivision operator given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{bmatrix}. \quad (65)$$

The terms \mathbf{S} , \mathbf{S}_{11} , \mathbf{S}_{12} , \mathbf{S}_{21} and \mathbf{S}_{22} are defined in [50] and \mathbf{S} is given in Eq. 61. To evaluate the sub-element Ω_1 , Ω_2 and Ω_3 in Fig. 4b, one needs to pick $2\kappa + 8$ control points out of the new $2\kappa + 17$ control point patch. A selection operator $\mathbf{D} + \kappa$ for sub-element Ω_k and $k = 1, 2, 3$ is used to select the necessary control points from \mathcal{P}_1 , that is

$$\mathcal{P}_{1,k} = \mathbf{D}_k \mathcal{P}_1. \quad (66)$$

Fig. 23 The weight distribution for computing different types of new control points



Then a surface point can be evaluated with the cubic spline basis functions as

$$\mathbf{x}(\xi) = \sum_{A=0}^{15} N_A(\xi) \mathbf{P}_A^{1,k}. \quad (67)$$

As shown in Fig. 4c, after successive subdivisions, the non-evaluable element can be limited to a negligible region.

Assume the target point has parametric coordinates $\xi = (\xi, \eta)$. One first determines how many subdivisions are required for this point by:

$$n = \lceil \min(-\log_2(\xi), -\log_2(\eta)) + 1 \rceil. \quad (68)$$

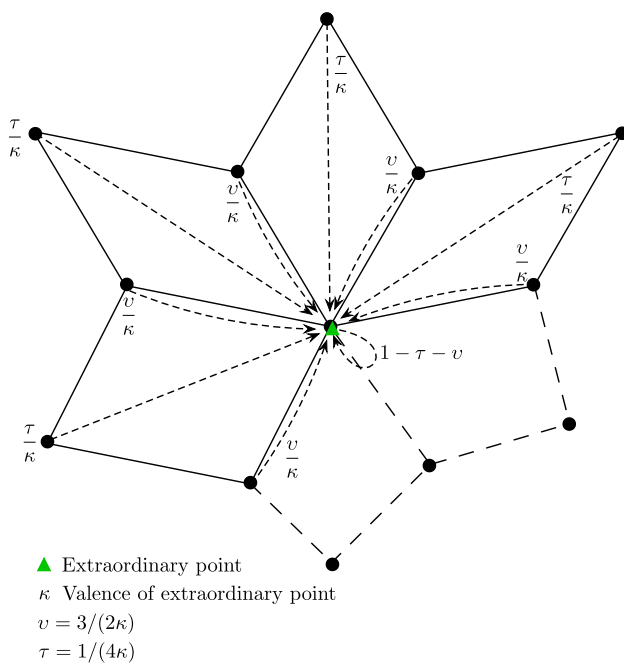


Fig. 24 Weight distributions for computing an extraordinary point with valence κ

The sub-element index k is determined as

$$k = \begin{cases} 1 & \text{if } \xi \in \left[\frac{1}{2^n}, \frac{1}{2^{n-1}} \right] \times \left[0, \frac{1}{2^n} \right], \\ 2 & \text{if } \xi \in \left[\frac{1}{2^n}, \frac{1}{2^{n-1}} \right] \times \left[\frac{1}{2^n}, \frac{1}{2^{n-1}} \right], \\ 3 & \text{if } \xi \in \left[0, \frac{1}{2^n} \right] \times \left[\frac{1}{2^n}, \frac{1}{2^{n-1}} \right]. \end{cases} \quad (69)$$

The surface point \mathbf{x} is located in the regular sub-element k after the n^{th} refinement. The patch for this element is picked with the selection operator \mathbf{D}_k as

$$\mathcal{P}_{n,k} = \mathbf{D}_k \mathcal{P}_n. \quad (70)$$

The enlarged set \mathcal{P}_n contains $2\kappa + 17$ control vertices, which is generated from the subdivision of \mathcal{P}_{n-1}^* as

$$\mathcal{P}_n = \mathbf{A} \mathcal{P}_{n-1}^*. \quad (71)$$

The set \mathcal{P}_{n-1}^* has $2\kappa + 8$ control vertices. It is successively refined from the initial set \mathcal{P}_0 as

$$\mathcal{P}_{n-1}^* = \bar{\mathbf{A}}^{n-1} \mathcal{P}_0, \quad (72)$$

where $\bar{\mathbf{A}}$ is a square matrix operator which subdivides the patch for computing the new patch for the irregular element, and is defined by

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{S}_{11} & \mathbf{S}_{12} \end{bmatrix}. \quad (73)$$

References

- Antoniotti P, Dedner A, Madhavan P, Stangalino S, Stinner B, Verani M (2014) High order discontinuous Galerkin methods on surfaces. [arXiv:1402.3428](https://arxiv.org/abs/1402.3428)
- Arndt D, Bangerth W, Clevenger TC, Davydov D, Fehling M, Garcia-Sanchez D, Harper G, Heister T, Heltai L, Kronbichler M, Kynch RM, Maier M, Pelteret JP, Turcksin B, Wells D (2019) The deal.II library, version 9.1. *J Numer Math* 27:203–213. <https://doi.org/10.1515/jnma-2019-0064>
- Arnold DN (1982) An interior penalty finite element method with discontinuous elements. *SIAM J Numer Anal* 19(4):742–760
- Autodesk, INC (2019) Maya. <https://autodesk.com/maya>
- Babuška I (1973) The finite element method with Lagrangian multipliers. *Numer Math* 20(3):179–192
- Bandara K, Cirak F (2018) Isogeometric shape optimisation of shell structures using multiresolution subdivision surfaces. *Comput Aided Des* 95:62–71
- Bandara K, Rüberg T, Cirak F (2016) Shape optimisation with multiresolution subdivision surfaces and immersed finite elements. *Comput Methods Appl Mech Eng* 300:510–539
- Bangerth W, Hartmann R, Kanschat G (2007) deal.II—a general purpose object oriented finite element library. *ACM Trans Math Softw* 33(4):24/1–24/27
- Bartezzaghi A, Dedè L, Quarteroni A (2015) Isogeometric analysis of high order partial differential equations on surfaces. *Comput Methods Appl Mech Eng* 295:446–469
- Bazilevs Y, Calo VM, Hughes TJR, Zhang Y (2008) Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Comput Mech* 43(1):3–37
- Bazilevs Y, Calo VM, Cottrell JA, Evans JA, Hughes TJR, Lipton S, Scott MA, Sederberg TW (2010) Isogeometric analysis using T-splines. *Comput Methods Appl Mech Eng* 199(5–8):229–263
- Borden MJ, Scott MA, Evans JA, Hughes TJR (2011) Isogeometric finite element data structures based on Bézier extraction of NURBS. *Int J Numer Methods Eng* 87(1–5):15–47
- Buchegger F, Jüttler B, Mantzaflaris A (2016) Adaptively refined multi-patch B-splines with enhanced smoothness. *Appl Math Comput* 272:159–172
- Catmull E, Clark J (1978) Recursively generated B-spline surfaces on arbitrary topological meshes. *Comput Aided Des* 10(6):350–355
- Chen L, Lu C, Lian H, Liu Z, Zhao W, Li S, Chen H, Borden SPA (2020) Acoustic topology optimization of sound absorbing materials directly from subdivision surfaces with isogeometric boundary element methods. *Comput Methods Appl Mech Eng* 362:112806
- Cirak F, Ortiz M (2001) Fully C^1 -conforming subdivision elements for finite deformation thin-shell analysis. *Int J Numer Methods Eng* 51:813–833
- Cirak F, Ortiz M, Schröder P (2000) Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *Int J Numer Methods Eng* 47(12):2039–2072
- Cottrell JA, Hughes TJR, Bazilevs Y (2009) Isogeometric analysis: toward integration of CAD and FEA. Wiley, Hoboken
- Dault D, Li J, Liu B, Zhao R, Tong Y, Shanker B (2015) Subdivision surfaces for electromagnetic integral equations. In: 2015 IEEE international symposium on antennas and propagation USNC/URSI national radio science meeting, pp 151–152

20. Dedè L, Quarteroni A (2015) Isogeometric analysis for second order partial differential equations on surfaces. *Comput Methods Appl Mech Eng* 284:807–834
21. Dedner A, Madhavan P (2016) Adaptive discontinuous Galerkin methods on surfaces. *Numer Math* 132(2):369–398
22. Dedner A, Madhavan P, Stinner B (2013) Analysis of the discontinuous Galerkin method for elliptic problems on surfaces. *IMA J Numer Anal* 33(3):952–973
23. Deng J, Chen F, Li X, Hu C, Tong W, Yang Z, Feng Y (2008) Polynomial splines over hierarchical T-meshes. *Graph Models* 70(4):76–86
24. Dokken T, Lyche T, Pettersen KF (2013) Polynomial splines over locally refined box-partitions. *Comput Aided Geom Des* 30(3):331–356
25. Dziuk G (1988) Finite elements for the Beltrami operator on arbitrary surfaces. In: *Partial differential equations and calculus of variations*. Springer, pp 142–155
26. Dziuk G, Elliott CM (2007) Surface finite elements for parabolic equations. *J Comput Math*, pp 385–407
27. Dziuk G, Elliott CM (2013) Finite element methods for surface PDEs. *Acta Numer* 22:289–396
28. Fernández-Méndez S, Huerta A (2004) Imposing essential boundary conditions in mesh-free methods. *Comput Methods Appl Mech Eng* 193(12–14):1257–1275
29. Forsey DR, Bartels RH (1988) Hierarchical B-spline refinement. *ACM Siggraph Comput Graph* 22(4):205–212
30. Giannelli C, Jüttler B, Speleers H (2012) THB-splines: the truncated basis for hierarchical splines. *Comput Aided Geom Des* 29(7):485–498
31. Gurtin ME, Jabbour ME (2002) Interface evolution in three dimensions with curvature-dependent energy and surface diffusion: interface-controlled evolution, phase transitions, epitaxial growth of elastic films. *Arch Ration Mech Anal* 163(3):171–208
32. Hansbo A, Hansbo P (2002) An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Comput Methods Appl Mech Eng* 191(47–48):5537–5552
33. Hughes TJR, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput Methods Appl Mech Eng* 194(39):4135–4195
34. Johannessen KA, Kvamsdal T, Dokken T (2014) Isogeometric analysis using LR B-splines. *Comput Methods Appl Mech Eng* 269:471–514
35. Jüttler B, Mantzaflaris A, Perl R, Rumpf M (2016) On numerical integration in isogeometric subdivision methods for PDEs on surfaces. *Comput Methods Appl Mech Eng* 302:131–146
36. Langer U, Mantzaflaris A, Moore SE, Touloupoulos I (2015) Multipatch discontinuous Galerkin isogeometric analysis. In: *Isogeometric analysis and applications 2014*. Springer, pp 1–32
37. Litke N, Levin A, Schröder P (2001) Fitting subdivision surfaces. In: *Proceedings of the conference on Visualization'01*. IEEE Computer Society, pp 319–324
38. Liu Z, Majeed M, Cirak F, Simpson RN (2018) Isogeometric FEM–BEM coupled structural-acoustic analysis of shells using subdivision surfaces. *Int J Numer Methods Eng* 113(9):1507–1530
39. Moës N, Béchet E, Tourbier M (2006) Imposing Dirichlet boundary conditions in the extended finite element method. *Int J Numer Methods Eng* 67(12):1641–1669
40. Neamtu M, Pfluger PR (1994) Degenerate polynomial patches of degree 4 and 5 used for geometrically smooth interpolation in 3. *Comput Aided Geom Des* 11(4):451–474
41. Nguyen T, Karčiauskas K, Peters J (2014) A comparative study of several classical, discrete differential and isogeometric methods for solving Poisson's equation on the disk. *Axioms* 3(2):280–299
42. Nguyen-Thanh N, Kiendl J, Nguyen-Xuan H, Wüchner R, Bletzinger KU, Bazilevs Y, Rabczuk T (2011) Rotation free isogeometric thin shell analysis using PHT-splines. *Comput Methods Appl Mech Eng* 200(47):3410–3424
43. Nitsche J (1971) Über ein variationsprinzip zur lösung von Dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. In: *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, Springer 36:9–15
44. Peters J (1991) Parametrizing singularly to enclose data points by a smooth parametric surface. In: *Proceedings of graphics interface*, vol 91
45. Peters J, Reif U (1998) Analysis of algorithms generalizing B-spline subdivision. *SIAM J Numer Anal* 35(2):728–748
46. Peterson AF (2005) Mapped vector basis functions for electromagnetic integral equations. *Synth Lect Comput Electromagn* 1(1):1–124
47. Reali A (2006) An isogeometric analysis approach for the study of structural vibrations. *J Earthq Eng* 10(1):1–30
48. Scott MA, Borden MJ, Verhoosel CV, Sederberg TW, Hughes TJR (2011) Isogeometric finite element data structures based on Bézier extraction of T-splines. *Int J Numer Methods Eng* 88(2):126–156
49. Sederberg TW, Zheng J, Bakenov A, Nasri A (2003) T-splines and T-NURCCs. In: *ACM transactions on graphics (TOG)*, vol 22. ACM, pp 477–484
50. Stam J (1998) Exact evaluation of Catmull–Clark subdivision surfaces at arbitrary parameter values. *SIGGRAPH Course Note* 98:395–404
51. Takacs T, Jüttler B (2012) H2 regularity properties of singular parameterizations in isogeometric analysis. *Graph Models* 74(6):361–372
52. Vuong A, Giannelli C, Jüttler B, Simeon B (2011) A hierarchical approach to adaptive local refinement in isogeometric analysis. *Comput Methods Appl Mech Eng* 200(49):3554–3567
53. Wawrzinek A, Polthier K (2016) Integration of generalized B-spline functions on Catmull–Clark surfaces at singularities. *Comput Aided Des* 78:60–70
54. Wei X, Zhang Y, Hughes TJR, Scott MA (2015) Truncated hierarchical Catmull–Clark subdivision with local refinement. *Comput Methods Appl Mech Eng* 291:1–20
55. Wei X, Zhang YJ, Hughes TJ, Scott MA (2016) Extended truncated hierarchical Catmull–Clark subdivision. *Comput Methods Appl Mech Eng* 299:316–336
56. Xiao X, Sabin M, Cirak F (2019) Interrogation of spline surfaces with application to isogeometric design and analysis of lattice-skin structures. *Comput Methods Appl Mech Eng* 351:928–950
57. Zhang Q, Sabin M, Cirak F (2018) Subdivision surfaces with isogeometric analysis adapted refinement weights. *Comput Aided Des* 102:104–114
58. Zienkiewicz O, Taylor RL (1997) The finite element patch test revisited a computer test for convergence, validation and error estimates. *Comput Methods Appl Mech Eng* 149(1–4):223–254